Spatially Coupled Sparse Regression Codes for Single- and Multi-user Communications

Kuan Hsieh

Robinson College
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

March 2021

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any work that has already been submitted, or, is being concurrently submitted for any degree or other qualification except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the Engineering Degree Committee of 65,000 words, including appendices, footnotes, tables and equations, but excluding the bibliography, and 150 figures.

Kuan Hsieh
March 2021

Spatially Coupled Sparse Regression Codes for Single- and Multi-user Communications

Kuan Hsieh

Abstract

Sparse regression codes (SPARCs) are a class of channel codes for efficient communication over the single-user additive white Gaussian noise (AWGN) channel at rates approaching the channel capacity. In a standard SPARC, codewords are sparse linear combinations of columns of an i.i.d. Gaussian design matrix, and the user message is encoded in the indices of those columns. Techniques such as power allocation and spatial coupling have been proposed to improve the performance of low-complexity iterative decoding algorithms such as approximate message passing (AMP).

In this thesis we investigate spatially coupled SPARCs, where the design matrix has a block-wise band-diagonal structure, and modulated SPARCs, which generalise standard SPARCs by introducing modulation to the encoding of user messages. We introduce a base matrix framework which provides a unified way to construct power allocated and spatially coupled design matrices, and propose AMP decoders for modulated SPARCs constructed using base matrices.

We prove that phase shift keying modulated and spatially coupled SPARCs with AMP decoding asymptotically achieve the capacity of the (complex) AWGN channel. We also show via numerical simulations that they can achieve lower error rates than standard coded modulation schemes at finite code lengths. A sliding window AMP decoder is proposed for spatially coupled SPARCs that significantly reduces the decoding latency and complexity.

We then investigate coding schemes based on random linear models and AMP decoding for the multi-user Gaussian multiple access channel in the asymptotic regime where the number of users grows linearly with the code length. For a fixed target error rate and message size per user (in bits), we obtain the exact trade-off between energy-per-bit and the user density achievable in the large system limit. We show that a coding scheme based on spatially coupled Gaussian matrices and AMP decoding achieves near-optimal trade-off for a large range of user densities. To the best of our knowledge, this is the first efficient coding scheme to do so in this multiple access regime. Moreover, the spatially coupled coding scheme has a practical interpretation: it can be viewed as block-wise time-division with overlap.

To my wife

Acknowledgments

I would like to thank my supervisor Ramji Venkataramanan for his detailed mentorship, constant support and encouragement, precise critiques and rigorous attitude towards research throughout my research journey.

I would like to thank Albert Guillén i Fàbregas and Giuseppe Caire for reading this thesis and for their thought provoking questions and constructive comments during my PhD viva.

I would like to thank my colleagues and faculty members in the Signal Processing and Communications Laboratory, for their inspiration, enthusiasm and great company. I would like to thank: Jossy Sayir for his contagious love for teaching and enthusiasm for coding; Albert Guillén i Fàbregas for his constant support and interesting discussions; Ioannis Kontoyiannis for being a great teacher and bringing researchers together regularly to discuss information theory; Adam Greig for the immense help he gave me in getting started in my project and for his patience in helping me with so many of my programming questions; Chenhao Li for his companionship throughout my research journey; Oliver Bonner, Fergal Cotter, Jos van der Westhuizen, Mahed Abroshan, Jiaming Liang, Jichi Deng, Amarjot Singh, Jacob Vorstrup, Ehsan Asadi, Parham Boroumand, Shirley Liu, Pablo Pascual Cobo, and the many other wonderful people in SigProc.

I would like to thank Cynthia Rush for engaging discussions and fruitful collaborations.

I would also like to thank Phill Richardson and Peter Grandi for their computing support, and Rachel Fogg, Laura Reed and Lina Zvaginyte-Bagociene for their administrative support.

I thank Sheng Yang for the research opportunity at ARM during my studies, and François Botman and Pranay Prabhat for their guidance during my internship.

I am grateful to Joshua Poon, John Chu, Tristan Wang, Henry Lu, Bryan Chang, Sarah Lee, Jonathan and Mary Woodruff, and many others in the church for being my family away from home, for their care, love and support in Christ.

I would like to thank my sister and brother-in-law Michelle (Hsuan) and Jonathan Holder, and my loving parents for their constant support and encouragement. Finally, I cannot thank my wife Sarah Beth Hsieh enough for her endless love, patience, support, and belief in me.

This work was generously funded by a Doctoral Training Partnership award from the Engineering and Physical Sciences Research Council. I'm grateful for their continued support during the COVID-19 pandemic. I'm also grateful to my supervisor Ramji Venkataramanan for his additional funding to support me during the final stages of my thesis write-up.

Preface

The work presented in Chapter 2 was done in collaboration with Ramji Venkataramanan (R.V.) and Cynthia Rush (C.R.), and has been accepted as part of a longer paper for publication in the IEEE Transactions on Information Theory; the preprint can be found on arXiv [1]. This work in its various stages has been published in several conference proceedings [2–4].

The work presented in Chapter 3 was done in collaboration with R.V. and has been accepted for publication in the IEEE Transactions on Information Theory; the preprint can be found on arXiv [5]. A shorter version of this work was published in a conference proceeding [6].

The work presented in Chapter 4 was done in collaboration with R.V. and C.R., and has been accepted to the 2021 IEEE International Symposium on Information Theory; the (extended) preprint can be found on arXiv [7].

Contents

1 Introduction **9**

1.1 AWGN channel . . . 9

1.2 Gaussian multiple access channel . . . 13

1.2.1 Finite-user setting . . . 14

1.2.2 Many-user setting . . . 14

1.3 Sparse regression codes . . . 18

1.3.1 Encoding . . . 18

1.3.2 Decoding . . . 20

1.4 Approximate message passing . . . 23

1.5 Spatial coupling . . . 29

1.5.1 Spatial coupling in LDPC codes . . . 29

1.5.2 Spatial coupling in compressed sensing . . . 33

1.5.3 Spatial coupling in sparse regression codes . . . 34

1.5.4 Other applications . . . 36

1.6 Structure of thesis . . . 36

1.7 Notation . . . 38

2 Spatially coupled sparse regression codes **40**

2.1 Spatially coupled SPARC construction . . . 41

2.2 AMP decoder . . . 44

2.2.1 The error performance of the AMP decoder . . . 47

2.3 Decoding progression according to state evolution . . . 48

2.3.1 Asymptotic analysis . . . 49

2.3.2 Non-asymptotic analysis . . . 53

2.4 Probability of excess section error rate . . . 55

2.5 Empirical performance . . . 56

2.5.1 Implementation details . . . 60

2.6 Sliding window AMP decoding . . . 62

2.6.1 Sliding window decoder description . . . 62

2.6.2 Empirical performance . . . 66
2.7 Proof of Proposition 2.3.1 . . . 68

3 Modulated sparse regression codes **72**
3.1 Introduction . . . 72
3.2 Modulated SPARC construction . . . 75
3.2.1 Power allocation and spatial coupling . . . 76
3.3 AMP decoder and state evolution . . . 77
3.3.1 AMP decoder . . . 77
3.4 Error performance analysis . . . 81
3.4.1 Error criteria . . . 81
3.4.2 Bounding the section error rate . . . 82
3.4.3 PSK-modulated SPARCs are capacity achieving . . . 86
3.5 Empirical error performance . . . 88
3.5.1 Decoding complexity . . . 90
3.6 Proofs . . . 91
3.6.1 Proof of Lemma 3.4.1 . . . 91
3.6.2 Proof of Proposition 3.4.1 . . . 93
3.6.3 Proof of Lemma 3.6.1 . . . 100
3.6.4 Proof of Proposition 3.4.2 . . . 104

4 Many-user Gaussian multiple access **109**
4.1 Random linear coding and AMP decoding . . . 110
4.1.1 Spatially coupled coding schemes . . . 110
4.1.2 AMP decoding and state evolution . . . 113
4.2 Asymptotic UER achieved by AMP decoding . . . 115
4.2.1 Potential function . . . 115
4.2.2 I.I.D. Gaussian matrices . . . 117
4.2.3 Spatially coupled Gaussian matrices . . . 122
4.2.4 Numerical results . . . 125
4.3 Large user payloads . . . 131
4.3.1 I.I.D. Gaussian codebooks . . . 133
4.3.2 Spatially coupled Gaussian codebooks . . . 135
4.3.3 Numerical results . . . 136
4.3.4 Implementation . . . 138
4.4 Complex Gaussian channel and coding schemes . . . 140
4.4.1 Theoretical results . . . 144
4.4.2 Numerical results . . . 146

5 Conclusions **149**

5.1 Future directions 150

Chapter 1

Introduction

This thesis investigates efficient capacity-achieving communication schemes for single- and multi-user channels. In Sections 1.1 and 1.2, we introduce the additive white Gaussian noise (AWGN) channel model for single-user communications, and the Gaussian multiple access channel model for multi-user communications. Furthermore, we describe the fundamental limits of communication in these channels. Then in Sections 1.3–1.5, we introduce the main frameworks and tools that we use to construct efficient capacity-achieving communication schemes: sparse regression codes, approximate message passing algorithms, and spatial coupling.

1.1 AWGN channel

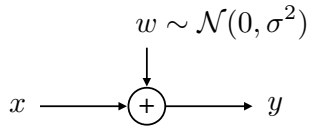


Figure 1.1: AWGN channel

The additive white Gaussian noise (AWGN) channel is depicted in Fig. 1.1. The channel generates output $y \in \mathbb{R}$ from input $x \in \mathbb{R}$ according to

$$y = x + w, \tag{1.1}$$

where w is drawn from a zero mean Gaussian distribution with variance σ^2, which we denote by $w \sim \mathcal{N}(0, \sigma^2)$. The AWGN channel is *memoryless* since the channel output y only depends on the current channel input x and not the previous inputs. The channel input has an average power constraint P: if $x_1, x_2, \ldots, x_n$ are transmitted over n uses of the channel, then it is required that

$$\frac{1}{n} \sum_{i=1}^{n} x_i^2 \leq P. \tag{1.2}$$

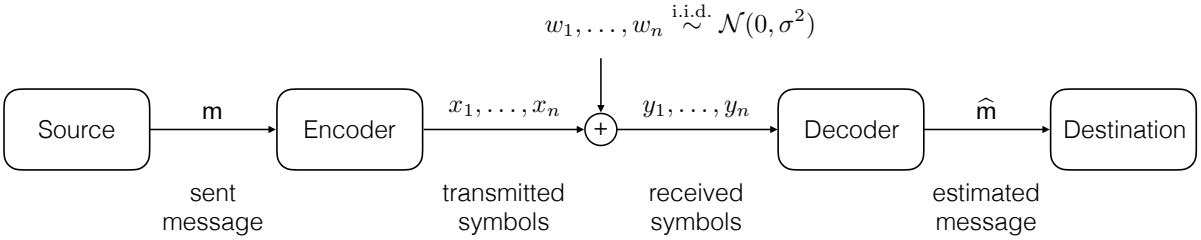


Figure 1.2: Communication across the AWGN channel

The signal-to-noise ratio of the AWGN channel is therefore P/σ^2.

This channel model is interesting and practically relevant because: (i) additive noise is commonly found in physical systems; (ii) Gaussian noise is a good approximation to the sum of many independent zero-mean random variables (which can represent different noise sources) due to the central limit theorem; (iii) Gaussian noise is the worst noise for a given variance (in the sense of differential entropy), so it provides a worst case bound; (iv) power constraints exist in real systems (e.g., use of battery); (v) it is easy to analyse. More complex models can be built on top of this.

Communication across the AWGN channel In order to communicate across the AWGN channel, one needs to *encode* information into channel input symbols, and *decode* channel output symbols back to meaningful information, see Fig. 1.2. The encoder maps a message m, chosen from a message set $\mathcal{M}$, into an input *codeword* $\boldsymbol{x} \in \mathbb{R}^n$. The decoder generates an estimate of the sent message denoted $\widehat{\mathsf{m}} \in \mathcal{M}$ from the channel output $\boldsymbol{y} \in \mathbb{R}^n$. The *probability of error* P_e is usually defined as either the maximum or average probability of decoding a message in error, i.e.,

$$P_e = \max_{\mathsf{m} \in \mathcal{M}} \mathbb{P}(\widehat{\mathsf{m}} \neq \mathsf{m}) \quad \text{or} \quad \sum_{\mathsf{m} \in \mathcal{M}} \mathbb{P}(\mathsf{m}) \mathbb{P}(\widehat{\mathsf{m}} \neq \mathsf{m}), \tag{1.3}$$

where $\mathbb{P}(\widehat{\mathsf{m}} \neq \mathsf{m})$ is the probability of decoding error given message m was transmitted.

Assuming all the messages are equally likely (which is a reasonable assumption when the messages come from a compressed source), then each message contains $\log_2 |\mathcal{M}|$ *bits* of information. Therefore, the *rate* of transmission denoted by R is given by

$$R = \frac{\log_2 |\mathcal{M}|}{n} \quad \text{bits/channel use.} \tag{1.4}$$

One aims to communicate at high rates with a low probability of error.

Shannon's channel coding theorem states that the *channel capacity* C (dependent on the channel parameters) is the tight upper bound on the rate at which reliable communication is achievable [8, 9]. In other words, for all $R < C$, there exists a code such that an arbitrarily small probability of error can be achieved. Conversely, for $R > C$, it is impossible to achieve an arbitrarily small probability of error. For the AWGN channel with average power constraint P

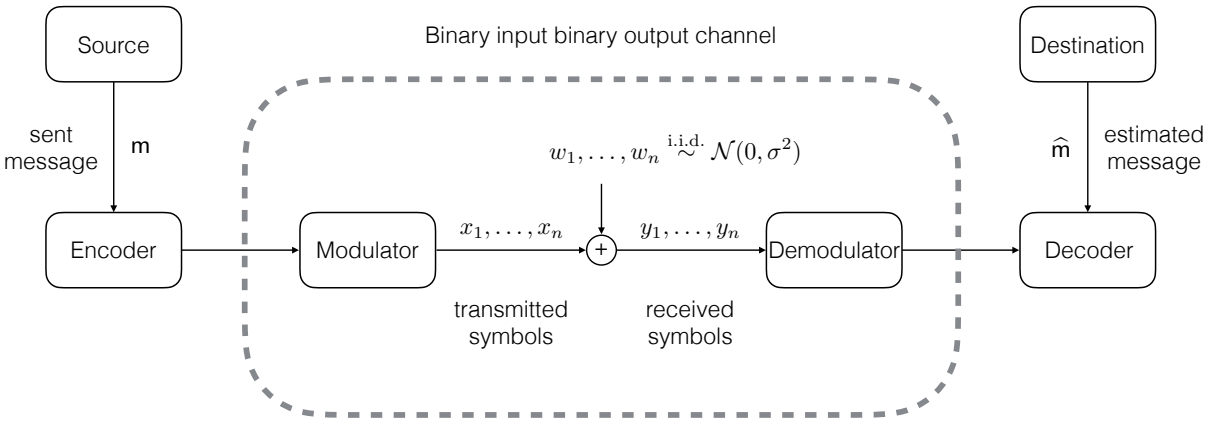


Figure 1.3: Coded modulation

and noise variance σ^2, the channel capacity is given by

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P}{\sigma^2} \right) \quad \text{bits/channel use.} \tag{1.5}$$

A key goal in information and coding theory is to communicate reliably across the AWGN channel at rates approaching the channel capacity with computationally efficient encoders and decoders.

In the standard forward proof of achievability, random codebooks with independent and identically distributed (i.i.d.) Gaussian entries are used and the code length n is taken to infinity [9]. However, decoding is computationally infeasible for this setup as the complexity grows exponentially with n (requires an exhaustive search over the whole codebook).

Since the channel coding theorem was stated in 1948, the channel coding community has been endeavouring to invent capacity achieving codes with feasible encoders and decoders. In the past few decades, much progress has been made to "close the gap to capacity", especially with in the the invention of Turbo codes [10], the rediscovery of Low-Density Party Check (LDPC) codes [11,12], and more recently the invention of polar codes [13] and spatially coupled LDPC codes [14, 15]. For an overview of the development of channel codes over the years, see [16].

Coded modulation Most practical coding schemes do not directly code for the AWGN channel. Instead, they are separated into two steps: *coding* and *modulation*, which is referred to as coded modulation [17–19].

The coding step includes an encoder, which maps messages $\mathsf{m} \in \mathcal{M}$ into binary sequences (or sequences of symbols from another finite set), and a decoder which reverses this operation. The modulation step includes a modulator and demodulator. The modulator uses standard schemes such as quadrature amplitude modulation (QAM) to map binary sequences onto a finite set

of symbols on a complex plane (known as constellations). These symbols are are then used to modulate the carrier waveform to be transmitted over the channel. The demodulator reverses the operation of the modulator. See Fig. 1.3 for the coded modulation system model.

For a fixed modulation scheme, the encoder and decoder views the modulator, channel and demodulator together as a channel with discrete inputs and outputs (dashed box in Fig. 1.3). State-of-the-art coded modulation schemes use binary error correcting codes such as Turbo, LDPC or polar codes for the coding step. Moreover, at the receiver, the demodulator passes soft-information to the decoder (posterior probabilities for each of the coded bits). The codes mentioned above have been shown to either come very close to, or provably achieve the channel capacity of *binary input* channels such as the binary erasure channel under practical decoders. However, when used together with popular modulation schemes such as QAM, they have good empirical performance but do not provably achieve the AWGN channel capacity.

Sparse regression codes One may ask if it were possible to step back from the coding/modulation divide and directly code for the AWGN channel (compare Figs. 1.2 and 1.3). Sparse regression codes (SPARCs) are a recent class of codes that generate real valued codewords to directly code for the AWGN channel, and are provably capacity achieving using various low-complexity[1] decoding algorithms [20–22].

In this thesis (Chapters 2 and 3) we consider SPARCs for AWGN channel coding under the efficient approximate message passing (AMP) decoder. We prove that certain generalisations of SPARCs can asymptotically achieve the channel capacity under AMP decoding. Furthermore, using these generalisations, we design SPARCs that achieve lower error rates than earlier SPARCs designs and standard coded modulation schemes, and have lower decoding complexity compared to earlier SPARCs. Introductions to SPARCs and the AMP algorithm are given in Sections 1.3 and 1.4, respectively.

Other methods We list here a few notable coding schemes for the AWGN channel which will not be discussed further in this thesis:

1. Trellis-coded modulation (TCM) [17] considers coding and modulation combined as a single entity for improved performance. TCM is based on the combination of trellis (convolutional) codes and constellation mappings via set partitioning.

2. Bit-interleaved coded modulation [19, 23] is a pragmatic approach to coded modulation where the encoder is a serial concatenation of a binary code, a bit interleaver, and a binary labelling function which maps bits to constellation symbols. Furthermore, the decoder

[1] By low-complexity, we mean the computational complexity is of the same order as a low order polynomial in the code length n.

receives soft information (e.g. bit-wise *a posteriori* probabilities) from the demodulator and the performance can be improved if the decoder is implemented iteratively.

3. Multilevel coding (MLC) using binary codes and multistage decoding (MSD) can achieve the AWGN channel capacity [24, 25].

4. Lattice codes are a class of structured codes that can achieve the AWGN channel capacity [26].

1.2 Gaussian multiple access channel

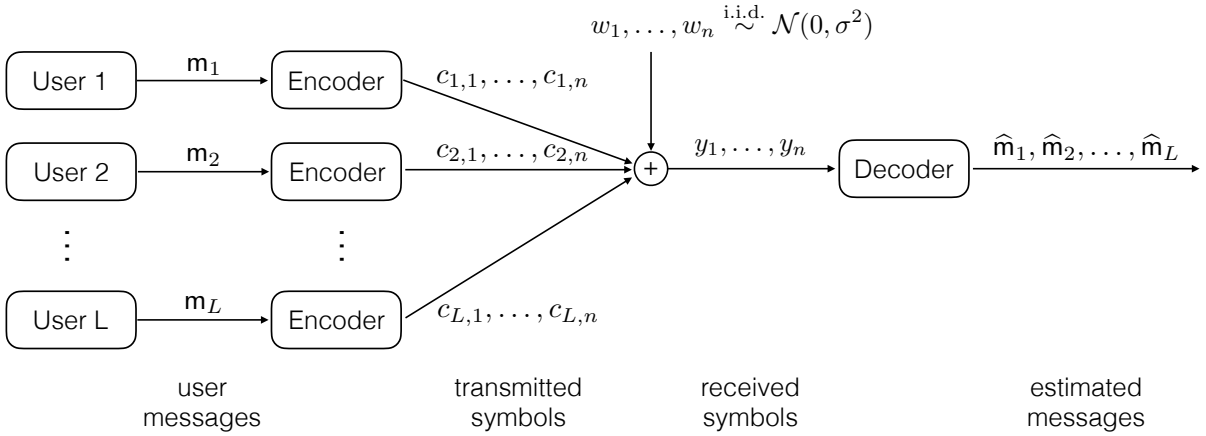


Figure 1.4: Communication across the Gaussian multiple access channel.

Often multiple users communicate simultaneously across a common channel to a single receiver. A useful model for this setting is the L-user Gaussian multiple access channel (MAC), where the channel output sequence/vector $\boldsymbol{y} \in \mathbb{R}^n$ is generated as

$$\boldsymbol{y} = \sum_{\ell=1}^{L} \boldsymbol{c}_\ell + \boldsymbol{w}, \tag{1.6}$$

where $\boldsymbol{c}_\ell \in \mathbb{R}^n$ is the codeword of user $\ell \in [L]$ and the entries of noise vector $\boldsymbol{w} \in \mathbb{R}^n$ are i.i.d. $\sim \mathcal{N}(0, \sigma^2)$, see Fig. 1.4. For a positive integer N, we use $[N]$ to denote the set $\{1, \ldots, N\}$. An example of the model (1.6) is a wireless network where multiple mobile devices communicate with a single base station over the same time or frequency block, and the electromagnetic waves containing the user information undergo constructive interference. A shortcoming of this channel model is that it assumes the users are coordinated and aligned in their use of the channel, i.e., synchronised.

This thesis concentrates on **symmetric** Gaussian MACs, where the message set $\mathcal{M}$ and the average power constraint of each user are the same. In this setting, the per user rate is given by

$R_{\text{user}} = \log_2 |\mathcal{M}|/n$ bits/channel use. In the rest of the thesis, the symmetric setting is assumed whenever MACs are mentioned.

Communication across Gaussian MACs involve trade-offs between the number of users, rate, signal-to-noise ratio, and the probability of decoding error. Furthermore, variables such as the probability of decoding error can be considered either on a per user basis, or on a joint (all user) basis. These trade-offs have been analysed under several asymptotic regimes in which the code length n is unbounded. We now discuss two such regimes.

1.2.1 Finite-user setting

In this setting, one considers a fixed number of users and aims to communicate at a high (per user) rates with a low *joint probability of error* (JPE):

$$\text{JPE} = \mathbb{P}((\widehat{\mathsf{m}}_1, \ldots, \widehat{\mathsf{m}}_L) \neq (\mathsf{m}_1, \ldots, \mathsf{m}_L)), \tag{1.7}$$

where each user's message, m_ℓ for $\ell \in [L]$, is chosen uniformly at random from the message set $\mathcal{M}$. This is the conventional setting considered in multi-user information theory, see [9, Chpt. 15] and [27, Chpt. 4].

Similar to the capacity of the (single-user) AWGN channel described in Section 1.1, the per user capacity of the (symmetric) Gaussian MAC, denoted by C_{user}, is the tight upper bound on the per-user rate R_{user} at which communication at arbitrarily low JPE is achievable. For the Gaussian MAC with noise variance σ^2 and average power constraint P, i.e.,

$$\|\boldsymbol{c}_\ell\|^2 \leq nP \quad \text{for } \ell \in [L], \tag{1.8}$$

the per user capacity is given by

$$C_{\text{user}} = \frac{1}{2L} \log_2 \left(1 + \frac{LP}{\sigma^2}\right) \quad \text{bits/channel use.} \tag{1.9}$$

When $L = 1$ we recover the AWGN channel capacity in (1.5). When L increases with the signal-to-noise ratio P/σ^2 held constant, the per user capacity decreases to 0. This suggests that for MACs with a large number of users, the constant user rate and (symbol) signal-to-noise ratio setting is not the one of interest. (Note that L tends to infinity after the code length n.)

1.2.2 Many-user setting

There has been a growing interest in the study of Gaussian MACs in the *many-user* setting, where the number of users L increases together with the code length n. This asymptotic setting was first introduced by Chen et al. in [28]. It was motivated by the need to model emerging digital communications scenarios where the number of devices in a network is expected to grow

significantly. These include scenarios introduced by Internet-of-Things (IoT) applications and massive machine type communications. Moreover, the trade-offs of interest in these applications often differ from those analysed in the finite-user setting. For example in wireless sensor networks, the number of bits transmitted by each sensor may be fixed instead of scaling with n (the latter happens when considering fixed per user rate), and there may be energy-per-bit requirements instead of power constraints on the user codeword symbols (see (1.8)). Therefore, the term "many-user" is used in the literature in a broad sense and individual works on many-user MACs may consider different trade-offs. In the literature, *many-access channels* and *massive multiple access* are often also used to refer to this multiple access setting [28, 29].

In the following sections, we discuss two lines of work on many-user MACs which analyse two different sets of trade-offs.

Effect of user scaling on capacity per unit energy

The *capacity per unit energy* $\tilde{C}_{\text{user}}$ is the maximum number of bits (payload) that each user can reliably transmit over the Gaussian MAC per unit energy, i.e., normalised by the squared norm of the user codewords $\|\boldsymbol{c}_\ell\|^2 = E$. In other words, for any payload per unit energy $\frac{\log_2|\mathcal{M}|}{E}$ less than $\tilde{C}_{\text{user}}$, there exists a code that can achieve an arbitrarily low JPE (defined in (1.7)).[2]

In [30], Ravi and Koch characterised the capacity per unit energy of Gaussian MACs in the many-user setting for different scalings of the number of users with the code length. Specifically, they showed that if the number of users L_n (a function of n) is of order strictly above $n/\log n$, then $\tilde{C}_{\text{user}} = 0$, i.e., no coding scheme can achieve a positive payload per unit energy. Conversely, if L_n is of order strictly below $n/\log n$, then the capacity per unit energy is equal to that of a single-user AWGN channel with the same noise variance as the Gaussian MAC, i.e., users can communicate interference free in this scaling regime.

In [31], the same authors considered the same setting as above, but defined the capacity per unit energy using the *per-user probability of error* (PUPE):

$$\text{PUPE} = \frac{1}{L}\sum_{\ell=1}^{L}\mathbb{P}(\widehat{\mathsf{m}}_\ell \neq \mathsf{m}_\ell), \tag{1.10}$$

instead of the joint probability of error (JPE) defined in (1.7). They showed that similar results hold with the transition threshold being at n instead of $n/\log n$: if L_n grows at least linearly with n, then $\tilde{C}_{\text{user}} = 0$. Otherwise, if L_n grows sublinearly in n, then $\tilde{C}_{\text{user}}$ is equal to that of a single-user AWGN channel. Both the above results consider vanishing probabilities of error (either JPE or PUPE $\to 0$ as $n \to \infty$).

In [31], the authors also considered non-vanishing probabilities of error. It was shown that the capacity per unit energy results for non-vanishing JPE are the same as that for vanishing

[2] Note that the inverse of the capacity per unit energy is the minimum energy-per-bit E_b required for reliably transmission.

JPE. That is, allowing JPE $\leq \epsilon$ for some fixed $\epsilon \in (0,1)$ that is independent of n does not increase the payload that can be transmitted per unit energy as $n \to \infty$.

However, the capacity per unit energy results for non-vanishing PUPE are different from those of vanishing PUPE. In particular, the authors argue that if the number of users grows linearly with the code length, a positive payload per unit energy is achievable if one allows for non-vanishing PUPE. Recall that $\tilde{C}_{\text{user}} = 0$ (vanishing PUPE) for linear orders of growth. Furthermore, in addition to requiring a non-vanishing PUPE to achieve positive payloads per unit energy, both the energy E and the payload size $\log_2 |\mathcal{M}|$ must be bounded as $n \to \infty$. In the following section, we describe results in this asymptotic scaling regime: *the number of users grows linearly with the code length, the user payload and energy are fixed, and a non-vanishing PUPE is considered.* This is the asymptotic regime that will be investigated in this thesis.

Tradeoff between user density and energy-per-bit in the linear scaling regime

In [32], Polyanskiy considered the Gaussian MAC in the asymptotic regime where the number of users L grows linearly with the code length n, i.e., $L = \mu n$ for some fixed *user density* μ, and the number of bits transmitted by each user (payload) is fixed and independent of n. In this asymptotic regime, Polyanskiy sought to characterise the optimal trade-offs between the user density μ, the user payload $\log_2 |\mathcal{M}|$, the per-user probability of error (PUPE) defined in (1.10), and the signal-to-noise ratio E_b/N_0. Here E_b is the energy-per-bit (the inverse of payload per unit energy) and $N_0/2 = \sigma^2$ is the noise spectral density per dimension. Note that the energy of the user codewords ($\|\boldsymbol{c}_\ell\|^2 = E$) are bounded since $E = E_b \log_2 |\mathcal{M}|$, and both E_b and $\log_2 |\mathcal{M}|$ are considered fixed.

In [32] and [33], Polyanskiy et al. obtained converse and achievability bounds on the minimum E_b/N_0 required to achieve a decoding error of PUPE $\leq \epsilon$ for a given $\epsilon \in (0,1)$, when the user density μ and user payload are fixed. The achievability bound was based on the coding scheme where users encode their messages with i.i.d. Gaussian codebooks, and messages are decoded with (joint) maximum likelihood (ML) decoding.

Fig. 1.5 shows an example of the results in [33]. We plot the converse (red) and achievability (blue) bounds on the minimum E_b/N_0 required to achieve a decoding error of PUPE $\leq 10^{-3}$ when the user payload is $\log_2 |\mathcal{M}| = 100$ bits. We observe that at this payload size and choice of maximum PUPE, the converse and achievability bounds match at user densities above approximately 0.008. We also observe an interesting behaviour in the these bounds: as E_b/N_0 increases, there is a sharp jump from not being able to communicate at any user density to being able to achieve a strictly positive user density. In this low user density region (vertical part of the curves), one can increase the user density up to the top of the vertical line without the need to increase the signal-to-noise ratio or suffer a higher PUPE, i.e., there is perfect multi-user interference cancellation. Recall from the previous section that interference free communications is also achievable if L grows sublinearly in n.

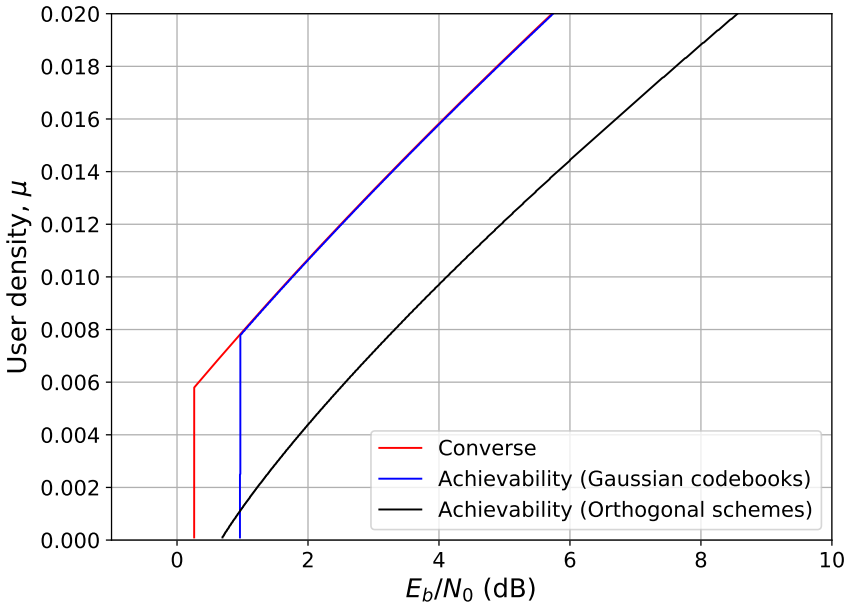


Figure 1.5: Asymptotic achievable regions in Gaussian MACs when the number of users L grows linearly with the code length n with fixed user density $\mu = L/n$. We plot the minimum E_b/N_0 required to achieve a decoding error of PUPE $\leq 10^{-3}$ when the user payload is $\log_2 |\mathcal{M}| = 100$ bits. (Same setup as [33, Fig. 1].)

In Fig. 1.5 we also plot the achievable region of orthogonal multiple access schemes (e.g. time division multiple access) for comparison (black), where the n channel uses are evenly divided among the L users. The achievable region of orthogonal schemes is obtained using the normal approximation to the (single-user) AWGN finite length error bound in [34] with the following AWGN parameters: code length $n/L = 1/\mu$, rate $\mu \log_2 |\mathcal{M}|$, and signal-to-noise ratio $2\mu \frac{E_b}{N_0} \log_2 |\mathcal{M}|$. We observe that the user density versus E_b/N_0 trade-off that can be achieved by orthogonal multiple access schemes is strictly suboptimal except at very small user densities, and the gap to the achievability bound (blue) is significant.

In this thesis (Chapter 4) we consider the Gaussian MAC in the same asymptotic setting: linear user scaling, finite payload, finite energy, and finite error probability. We analyse coding schemes based on random linear models (which include i.i.d. Gaussian codebooks) and efficient approximate message passing (AMP) decoding, and derive the exact asymptotic achievability regions of these schemes. We find that the asymptotic achievability of a coding scheme based on *spatially coupled* Gaussian matrices and AMP decoding exceeds that suggested by the achievability bound in [33] and nearly matches the converse bound for a large range of user densities. To the best of our knowledge, this is the first efficient coding scheme to do so in this MAC regime. The spatially coupled scheme can be interpreted as generalised time-sharing: the coupling structure specifies which users are active during each channel use. Introductions to the AMP algorithm and spatial coupling are given in Sections 1.4 and 1.5, respectively.

Other works on many-user MACs

The recent papers [35, 36] study the fundamental trade-offs in the quasi-static fading MAC in same asymptotic regime as [32, 33], where the number of users grows linearly with the code length.

Several works on multiple access in many-user setting consider *random* multiple access [28, 32, 37], *unsourced* multiple access, or a combination of both [32, 38–40]. Random (or uncoordinated) multiple access is where only a subset of all users are actively transmitting. The user activity is sporadic and the active users need to be identified by the receiver. Moreover, the number of active users may grow with the code length at a different rate compared to the total number of users. This can model the sporadic nature of some event-driven IoT applications. Unsourced multiple access is where all users use the same codebook and hence user messages are decoded up to a permutation, i.e., without regard to who the sender is. The combination of the two, which is called unsourced random access, can be used to model multiple access in IoT applications where the population statistics is of interest rather than any individual's data, perhaps for privacy preserving reasons. A review of the recent developments in many-user MACs can be found in [29].

1.3 Sparse regression codes

Sparse superposition codes, or sparse regression codes (SPARCs), are a recent class of codes introduced by Joseph and Barron for reliable communication over the (single-user) AWGN channel (1.1) [20,21]. Since SPARCs were introduced, they have been generalised for communication over general (single-user) memoryless channels [41, 42], and applied to lossy compression [43], Gaussian multi-terminal source and channel coding problems [44], and unsourced random access [39, 40]. The recently published monograph on SPARCs [45] provides an overview of the research that have been done in SPARCs. In this thesis, we focus on SPARCs for AWGN channel coding.

1.3.1 Encoding

A standard SPARC is defined by a random *design matrix* $\boldsymbol{A}$ of dimensions $n \times LM$ whose entries are i.i.d. zero mean Gaussian. Here n is the code length and L, M are integers whose significance will be explained later. As shown in Fig. 1.6, the design matrix $\boldsymbol{A}$ can be viewed as having L sections with M columns each. Codewords are generated by the linear combination of L columns of $\boldsymbol{A}$, one column from each section. This can be represented as a matrix-vector multiplication $\boldsymbol{A}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^{LM}$ is a *message vector* with exactly one non-zero entry in each of its L sections. The corresponding non-zero values are fixed a priori and denoted by $a_1, \ldots, a_L$ as shown in Fig. 1.6. Note that SPARCs are not linear codes as the sum of two codewords is

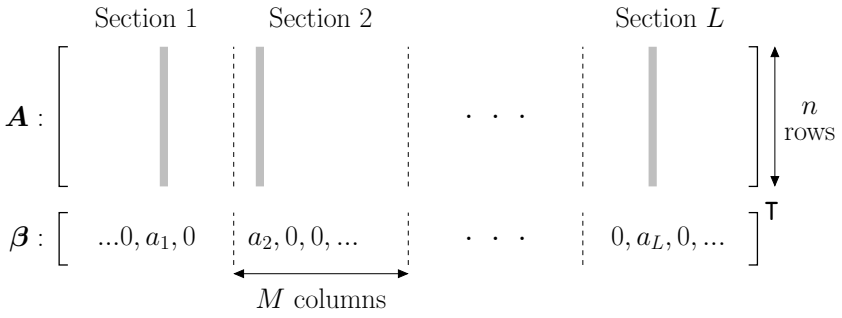


Figure 1.6: SPARC codewords are of the form $\boldsymbol{A\beta}$, where $\boldsymbol{A}$ is an $n \times LM$ design matrix and $\boldsymbol{\beta}$ is an $LM \times 1$ message vector with one non-zero entry in each of its L sections. The non-zero values $a_1, \ldots, a_L$ are fixed a priori.

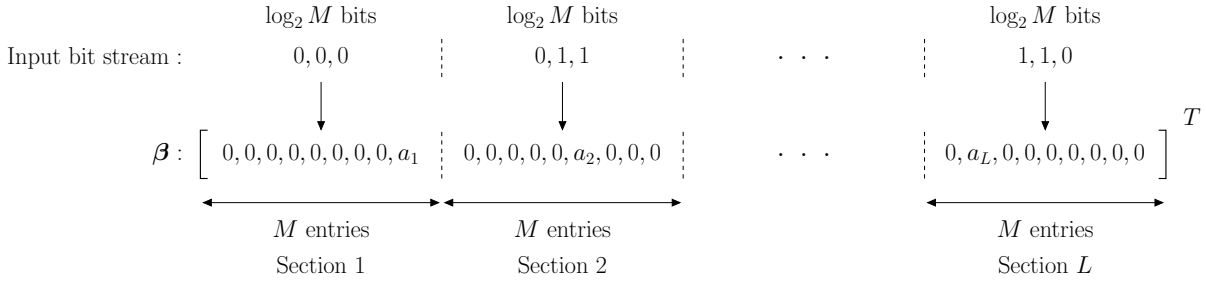


Figure 1.7: SPARC encoding example with $M = 8$.

not necessarily a codeword. (We intentionally use L to denote both the number of users in a Gaussian MAC (Section 1.2) and the number of sections in a SPARC. The connections between the Gaussian MAC model and the SPARC construction are described in Chapter 4.)

The message to be transmitted is indexed by the *locations* of the non-zeros in the message vector $\boldsymbol{\beta}$. Since each section of the L sections of $\boldsymbol{\beta}$ has M entries, each section encodes $\log_2 M$ bits, and the rate of the code is given by

$$R = \frac{L \log_2 M}{n} \quad \text{bits/channel use.} \tag{1.11}$$

A graphical illustration of the encoding procedure is given in Fig. 1.7, where we use the decimal equivalent of each segment of $\log_2 M$ bits to determine the location of the single non-zero entry among M corresponding entries of the message vector. Note that the message vector is *sparse*, i.e., most entries in $\boldsymbol{\beta}$ are zero.

Power allocation The design of the non-zero values in the message vector $\{a_1, \ldots, a_L\}$ is denoted the *power allocation*. Power allocation is crucial to the performance of SPARCs when efficient iterative decoders are used [21, 22, 46, 47]. A commonly used design for theoretical analysis is the exponentially decaying power allocation:

$$a_\ell \propto 2^{-2C\ell/L} \quad \text{for } \ell \in [L], \tag{1.12}$$

where C is the Shannon capacity of the AWGN channel given in (1.5). The variance of the Gaussian entries of $\boldsymbol{A}$ and the power allocation $\{a_1, \ldots, a_L\}$ are chosen such that codewords satisfy the average power constraint in (1.2) in expectation, i.e., $\mathbb{E}[\|\boldsymbol{A}\boldsymbol{\beta}\|^2] \leq nP$. The intuition for using this exponentially decaying power allocation for iterative decoding is that the sections with higher power (larger values of a_ℓ) decode in the earlier iterations; once decoded, the interference from these high power sections are removed, which makes lower power sections easier to decode in later iterations.

1.3.2 Decoding

Using the AWGN channel model in (1.1), the channel output $\boldsymbol{y} \in \mathbb{R}^n$ can be represented as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\beta} + \boldsymbol{w}, \tag{1.13}$$

where the noise vector $\boldsymbol{w} \in \mathbb{R}^n$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. The decoder aims to recover the message vector $\boldsymbol{\beta}$ given the channel output $\boldsymbol{y}$. The design matrix $\boldsymbol{A}$, the power allocation $\{a_1, \ldots, a_L\}$, and the channel noise variance σ^2 are known to the decoder. Notice that this is similar to the compressed sensing signal recovery problem where one aims to recover a sparse (in some known basis) vector $\boldsymbol{\beta}$ from random linear measurements $\boldsymbol{y}$ [48–50].[3]

Error performance criterion A common performance measure of SPARC decoders is the *section error rate* (SER), which is the fraction of sections decoded wrongly, or equivalently, the fraction of the locations of the non-zero entries estimated in error. It is defined as

$$\text{SER} := \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\left\{\hat{\boldsymbol{\beta}}_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)}\right\}, \tag{1.14}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, $\boldsymbol{\beta}_{\text{sec}(\ell)} \in \mathbb{R}^M$ is the ℓ-th section of the message vector, and $\hat{\boldsymbol{\beta}}_{\text{sec}(\ell)}$ is the decoder's estimate of that section. For several decoders (which are listed later), prior works have obtained bounds on the probability of excess section error rate, i.e., the probability of the event $\{\text{SER} > \epsilon\}$ for some $\epsilon \in (0, 1)$.

Another performance measure is the *bit error rate* (BER). Recall that $\log_2 M$ bits determine the location of the non-zero entry in a section of the message vector $\boldsymbol{\beta}$. Assuming that the probability of estimating the location of the non-zero entry incorrectly is uniform across the M possible locations, if the location is estimated incorrectly, then on average half of the $\log_2 M$ bits will be decoded in error due to the uniform mapping of location to bits. Therefore, for a

[3] Compared to compressed sensing, the $\boldsymbol{\beta}$ vector for SPARCs has section-wise i.i.d. entries (one non-zero entry in each section whose value is known to the decoder, only the location is unknown) whereas compressed sensing usually assumes i.i.d. entries. Moreover, in the asymptotic analysis of compressed sensing, one often assumes that the measurement ratio $\frac{\dim(\boldsymbol{y})}{\dim(\boldsymbol{\beta})}$ converges to a constant, whereas in the asymptotic analysis of SPARCs, the measurement ratio tends to 0. See Section 1.4 for more details on compressed sensing.

SPARC with a large number of sections, the bit error rate will be close to half the section error rate.

Optimal decoder SPARCs with "flat" power allocation ($a_1 = a_2 = \ldots = a_L$) were analysed under the optimal/maximum likelihood (ML) decoder in [20]. The decoder is given by

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}' \in \mathcal{B}_{L,M}}{\arg\min} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}'\|_2^2, \tag{1.15}$$

where $\mathcal{B}_{L,M}$ is the set of length LM vectors with L sections of M entries each, and with a single non-zero entry equal to a_1 in each section. For rates $R < C$, the ML decoder was shown in [20] to have error probability decaying exponentially in the code length n. In particular, the probability of excess section error rate is bounded as follows for any $\epsilon \in (0, 1)$:

$$\mathbb{P}(\text{SER} > \epsilon) \leq e^{-\kappa n \min(\epsilon, (C-R)^2)}, \tag{1.16}$$

where κ is a universal positive constant. This result was extended to SPARCs with design matrices that have i.i.d. Bernoulli ± 1 entries in [51, 52]. Is not computationally feasible to implement the ML decoder for large values of (L, M), hence, several low complexity iterative decoders have been proposed.

Low complexity decoding schemes Here we list several low complexity iterative decoders that have been proposed for SPARCs. The results that we discuss are all given for SPARCs with exponentially decaying power allocations (see (1.12)).

1. **Adaptive successive hard-decision decoder** For any rate $R < C$, the probability of excess section error rate of this decoder decays exponentially in $n/\log n$ [21].

2. **Adaptive successive soft-decision decoder** For any rate $R < C$, the probability of excess section error rate of this decoder decays exponentially in $n/(\log n)^{2T^*+1}$, where T^* is the number of iterations run by the decoder [46, 53].

3. **Approximate message passing decoder** This decoder has been proposed and analysed in several works [22, 54–57]. For any rate $R < C$, the excess section error rate of the AMP decoder decays exponentially in $n/(\log n)^{2T^*+1}$, where T^* is the number of iterations required for successful decoding and is inversely proportional to $\log(C/R)$ [57].

Error performance Although the error exponents for the above three decoders are similar, their empirical performance differs. In particular, the adaptive successive hard-decision decoder has high SER at rates near capacity for practical code lengths [58]. To best of the author's knowledge, the latter two decoder's error performance have not been directly compared.

Computational complexity The computational complexity and memory requirement of the adaptive successive hard-decision decoder and the AMP decoder are both of order $O(nLM)$ when Gaussian design matrices used. The complexity is dominated by matrix-vector multiplications with the design matrix $\boldsymbol{A} \in \mathbb{R}^{n \times LM}$ and the memory requirement by the need to store the design matrix. The adaptive successive soft-decision decoder has higher decoding complexity as it requires a Cholesky decomposition computation at each iteration. In [22] and [56], the authors proposed to construct design matrices $\boldsymbol{A}$ by uniformly sampling rows from a Hadamard or Fourier matrix, similar to what is often done in the compressed sensing literature [50]. This can reduce the computational complexity to $O(LM \log(LM))$ by using the Fast Walsh-Hadamard Transform [59] or the Fast Fourier Transform [60]. Furthermore, only the index of the rows that were sampled need to be stored in memory, which is of order $O(n)$.

Hardware implementations There has been some research done in the design of dedicated hardware for SPARC encoding and decoding for improved efficiency. In [61], the error performance of SPARCs under the AMP decoders introduced in [22] and [56] were evaluated under finite precision and finite code length. In [62], the same authors proposed the first SPARC encoder and decoder architectures.

Improving error performance Two approaches have been proposed to improve the finite length error performance of SPARCs under AMP decoding. In [22, 47], the authors proposed ways to optimise the power allocation, and in [55,56], the authors proposed to use the *spatial coupling* technique to construct the design matrix $\boldsymbol{A}$. For rates away from capacity and at practical code lengths, these methods showed orders of magnitude improvement in the SER compared to using the exponentially decaying power allocation defined in (1.12). The spatial coupling technique will be introduced in Section 1.5 and its application to SPARCs in Chapter 2.

In this thesis (Chapters 2 and 3) we devise a unified framework to analyse both power allocated and spatially coupled SPARCs. Furthermore, we generalise the SPARC construction so that information is not only encoded in the locations of the non-zero entries of the message vector $\boldsymbol{\beta}$, but also in the non-zero values that they take. These generalised SPARCs are named *modulated* SPARCs as the non-zero values are chosen from a digital modulation scheme such as phase-shift keying (PSK). We analyse the decoding progression of spatially coupled and PSK modulated SPARCs under AMP decoding, and prove that they are asymptotically capacity achieving. We also show via simulations that they can achieve lower error rates than existing SPARCs constructions and also standard coded modulation schemes at finite code lengths.

1.4 Approximate message passing

Approximate Message Passing (AMP) refers to a class of algorithms that are Gaussian/quadratic approximations to loopy belief propagation/message passing algorithms on dense factor graphs for certain high-dimensional problems, e.g., compressed sensing, generalised linear models, and low-rank matrix estimation [63–66]. The AMP algorithm is closely related to the "approximate belief propagation" algorithms used in earlier works for code-division multiple-access (CDMA) multi-user detection problems [67–70].

In this thesis we focus on using AMP as a decoding algorithm for coding schemes based on random linear models. In this section we introduce AMP in the context of the compressed sensing problem, which is a particular application of random linear models.

Compressed sensing In compressed sensing [48–50], one aims to recover a signal vector $\boldsymbol{\beta}_0 \in \mathbb{R}^n$ from random linear measurements $y_i = \boldsymbol{a}_i \cdot \boldsymbol{\beta}_0$ for $i = 1, \dots, m$, where the $\boldsymbol{a}_i$'s are random vectors, $\boldsymbol{x} \cdot \boldsymbol{y}$ represents the dot product between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, and the number of measurements m is less than the signal dimension n. Signal recovery is possible because the recovery algorithm takes advantage of a priori knowledge of the signal structure, such as *sparsity* (only a subset of entries of $\boldsymbol{\beta}_0$ are non-zero), or sparsity is some known basis (e.g., Wavelet or Fourier). Compressed sensing in additive Gaussian noise is often represented in the following vector form:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\beta}_0 + \boldsymbol{w}, \tag{1.17}$$

where the sensing (or measurement) matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has $\boldsymbol{a}_1, \dots, \boldsymbol{a}_m$ as its rows and the noise vector $\boldsymbol{w} \in \mathbb{R}^m$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries.

Compressed sensing is often analysed in the large system limit where m, n both tend to infinity with the measurement ratio $\delta = m/n$ converging to a constant in $(0, \infty)$. Furthermore, i.i.d. Gaussian sensing matrices and i.i.d. signal vectors are often assumed, i.e., $A_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/m)$ and $\beta_{0,j} \overset{\text{i.i.d.}}{\sim} p_{\beta_0}$ [71–74]. We now introduce the AMP signal reconstruction algorithm in such a setting.

AMP signal reconstruction for i.i.d. Gaussian $\boldsymbol{A}$ A compressed sensing signal reconstruction algorithm aims to accurately recover the signal vector $\boldsymbol{\beta}_0$ from noisy measurements $\boldsymbol{y}$ given the sensing matrix $\boldsymbol{A}$. The AMP algorithm iteratively generates signal vector estimates $\boldsymbol{\beta}^t$ at iterations $t = 0, 1, 2, \dots$ as follows: initialise $\boldsymbol{\beta}^0$ to the all-zero vector, and for $t \geq 0$ compute

$$\begin{aligned} \boldsymbol{z}^t &= \boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}^t + \frac{1}{\delta}\boldsymbol{z}^{t-1} \left\langle \eta'_{t-1}(\boldsymbol{\beta}^{t-1} + \boldsymbol{A}^* \boldsymbol{z}^{t-1}) \right\rangle, \\ \boldsymbol{\beta}^{t+1} &= \eta_t \left(\boldsymbol{\beta}^t + \boldsymbol{A}^* \boldsymbol{z}^t \right). \end{aligned} \tag{1.18}$$

Here $\boldsymbol{A}^*$ is the transpose of $\boldsymbol{A}$, variables with negative iteration indices are set to zero, $\eta_t(\cdot)$'s are scalar denoising functions (applied entry-wise) that we discuss later, $\eta_t'(s) = \frac{\partial}{\partial s}\eta_t(s)$, and for a vector $\boldsymbol{x} = [x_1, ..., x_n]$, $\langle \boldsymbol{x} \rangle := \sum_{i=1}^n x_i/n$. There is a vast literature on the history of AMP algorithms and how they can be obtained by making approximations to sum-product (or min-sum) message passing, a few references are [64,75–78]. Before we discuss the $\eta_t(\cdot)$ denoising functions, we need to introduce *state evolution*.

State evolution State evolution (SE) is a deterministic recursion that tracks the per-iteration performance of the AMP algorithm in the large system limit ($m, n \to \infty$ with m/n converging to a constant). It is similar to the density evolution recursion, which tracks the per-iteration erasure probability of the belief propagation decoder for sparse graph error correcting codes in the limit of large code length [79]. In particular, state evolution tracks the evolution of a mean squared error (MSE) term ψ_t and an effective noise variance term τ_t across iterations as follows: initialise $\psi_0 = \mathbb{E}[\beta^2]$, and for $t \geq 0$ compute

$$\begin{aligned} \tau_t &= \sigma^2 + \frac{1}{\delta}\psi_t, \\ \psi_{t+1} &= \mathbb{E}\left\{ [\eta_t(\beta + \sqrt{\tau_t}Z) - \beta]^2 \right\}, \end{aligned} \tag{1.19}$$

where σ^2 is the measurement noise variance, and the expectation is taken over the random variable $\beta \sim p_{\beta_0}$ and $Z \sim \mathcal{N}(0,1)$ independent of β.

For scalar functions $\eta_t(\cdot)$ which are Lipschitz continuous and signal distributions p_{β_0} that satisfy certain bounded moment conditions, [76, Thm. 1] shows that the per-iteration MSE of the AMP estimate is accurately tracked by ψ_t in the large system limit. That is, for $t \geq 0$, almost surely[4]

$$\lim_{n\to\infty} \frac{1}{n} \left\| \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}_0 \right\|^2 = \mathbb{E}\left\{ [\eta_t(\beta + \sqrt{\tau_t}Z) - \beta]^2 \right\} = \psi_{t+1}. \tag{1.20}$$

Interpretation of the AMP decoder At each iteration the AMP algorithm first produces a modified residual term $\boldsymbol{z}^t$, which consists of a residual term $\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}^t$ and an additional "Onsager" correction term which is a function of the previous modified residual $\boldsymbol{z}^{t-1}$.[5] The AMP algorithm then produces an effective observation term $\boldsymbol{s}^t = \boldsymbol{\beta}^t + \boldsymbol{A}^*\boldsymbol{z}^t$ which is the input to the denoising function $\eta_t(\cdot)$. The key intuition behind the AMP algorithm is that the entries of $\boldsymbol{s}^t$ can be seen as noisy versions of the entries of the signal $\boldsymbol{\beta}_0$. In particular, for index $j \in [n]$, s_j^t approximately distributed as $\beta_{0,j} + \sqrt{\tau_t}Z$, where Z is a standard Gaussian independent of

[4] The result of [76, Thm. 1] holds for more general functions on $\boldsymbol{\beta}^{t+1}$ and $\boldsymbol{\beta}_0$ (other than the squared error function), and only requires the empirical distribution of $\boldsymbol{\beta}_0$ to converge weakly to a probability measure p_{β_0} on $\mathbb{R}$ with certain bounded moment conditions.

[5] The correction term is similar to momentum terms often found in accelerated first order optimisation algorithms, e.g., Nesterov's accelerated gradient method. However, the correction term is based on the residual term $\boldsymbol{z}^t$ and not the signal term $\boldsymbol{\beta}^t$ as done in momentum methods. Furthermore, the correction term has a specific characterisation (no tuning parameters) which differentiates it from momentum methods.

$\beta_{0,j}$, and τ_t is the state evolution parameter given in (1.19). Therefore, the role of the $\eta_t(\cdot)$ function is to estimate the signal entries $\{\beta_{0,j}\}_{j\in[n]}$ in Gaussian noise. The Onsager correction term is crucial to the above distributional property of $\boldsymbol{s}^t$. For more intuition on its role in the AMP algorithm, see [76, Sec. I.C].

If the prior distribution of the signal p_{β_0} is known (e.g., in communication problems where an engineer designs the "signal"), then the estimator $\eta_t(\cdot)$ that minimises the MSE is the conditional expectation, i.e.,

$$\eta_t(s) = \mathbb{E}\left[\beta \mid \beta + \sqrt{\tau_t} Z = s\right]. \tag{1.21}$$

Example: sparsity In compressed sensing applications, the signal $\boldsymbol{\beta}_0$ is often assumed to be sparse and the signal recovery algorithm may not know the signal generating distribution (or there may not be one). Let us assume that the entries of $\boldsymbol{\beta}_0$ are drawn i.i.d. from p_{β_0}, and that p_{β_0} belongs to the class of distributions with sparsity ratio ϵ, i.e., $\mathbb{P}(\beta_0 \neq 0) = \epsilon$. The *soft-thresholding function* is near-optimal for estimating such sparse signals in Gaussian noise in a minimax sense, i.e., it achieves near minimum MSE for the worst case distribution in the class of distributions with sparsity ratio ϵ [80, 81]. The soft-thresholding function $S(s;\theta_t)$ is defined as follows:

$$S(s;\theta_t) = \begin{cases} s - \theta_t & \text{if } s > \theta_t, \\ 0 & \text{if } |s| \leq \theta_t, \\ s + \theta_t & \text{if } s < \theta_t. \end{cases} \tag{1.22}$$

Using $\eta_t(s) = S(s;\theta_t)$ in the AMP algorithm (1.18), we obtain the soft-thresholding AMP algorithm:

$$\begin{aligned} \boldsymbol{z}^t &= \boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}^t + \frac{\|\boldsymbol{\beta}^t\|_0}{m}\boldsymbol{z}^{t-1}, \\ \boldsymbol{\beta}^{t+1} &= S\left(\boldsymbol{\beta}^t + \boldsymbol{A}^*\boldsymbol{z}^t; \theta_t\right). \end{aligned} \tag{1.23}$$

Here $\|\boldsymbol{x}\|_0$ denotes the ℓ_0-pseudo-norm of vector $\boldsymbol{x}$, i.e., the number of its non-zero entries. A suitable choice for the threshold parameters $\{\theta_t\}_{t\geq 0}$ is to be proportional to the standard deviation of the effective noise, i.e., $\theta_t = \alpha\sqrt{\tau_t}$ for some tuning parameter α. A specific prescription for the choice of α (that is dependent on the sparsity ratio ϵ) is given in [77, Sec. 3].

It was proved in [72] that the asymptotic MSE of the soft-thresholding AMP algorithm (1.23) coincides with that of the popular LASSO estimator for compressed sensing [82]:

$$\hat{\boldsymbol{\beta}}_{\text{LASSO}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{1.24}$$

where the ℓ_1 penalty term favours sparse solutions and $\lambda > 0$ is a tuning parameter. This connection opened up a new way to analyse the LASSO, e.g., the noise-sensitivity of the LASSO

estimates [71], and also provided an algorithm to solve it (for i.i.d. Gaussian sensing matrices).

Potential function analysis and connections to MMSE estimation In addition to being able to analyse the per-iteration error of the AMP algorithm via state evolution, the error of the AMP algorithm at convergence (considering $t \to \infty$) can be analysed via the *potential function* method [83].

Consider the linear model (1.17) and the AMP algorithm (1.18) with the denoising function $\eta_t(\cdot)$ chosen to be the minimum mean squared error (MMSE) estimator in (1.21). For this setting, the following potential function has been proposed (up to additive constants) in many works [54, 56, 74, 78, 84]:[6]

$$\mathcal{F}(\delta, \sigma^2, \psi) = I\left(\beta; \beta + \sqrt{(\sigma^2 + \psi/\delta)}Z\right) + \frac{\delta}{2}\left[\ln\left(\frac{\sigma^2 + \psi/\delta}{\sigma^2}\right) - \frac{\psi/\delta}{\sigma^2 + \psi/\delta}\right], \tag{1.25}$$

where $\delta = m/n$ is the measurement ratio, σ^2 is the measurement noise variance, and ψ is an MSE term similar to that in state evolution (1.19). The mutual information term $I(\cdot;\cdot)$ is calculated using $\beta \sim p_{\beta_0}$ and $Z \sim \mathcal{N}(0,1)$ independent of β. The stationary points of this function with respect to ψ correspond to the fixed points of state evolution recursion (1.19). (One can verify this via the I-MMSE relationship [85]). Hence, analysing the stationary points of the potential function gives insight into asymptotic MSE achieved by the AMP algorithm at convergence. In particular, when the stationary point is unique, the asymptotic MSE achieved by the AMP ($\lim_{n\to\infty} \|\boldsymbol{\beta}^t - \boldsymbol{\beta}_0\|^2/n$) converges to the ψ value at the stationary point, which has been proven to be the minimum achievable MSE [74, 76, 84]. The same works also show that when the stationary point is not unique and the signal distribution p_{β_0} satisfies certain conditions, the ψ value at the global minimum of the potential function corresponds to the minimum achievable MSE, and the largest ψ value at a stationary point corresponds to asymptotic MSE achieved by the AMP algorithm.

Potential functions have also been used to analyse sparse graphs codes and belief propagation decoding, where the stationary points of the *Bethe free energy* correspond to the fixed points of density evolution [83, 86, 87]. In Section 1.5 we will see how potential functions are a key ingredient in analysing the performance gains obtained by spatial coupling.

Generalisations and other applications

Denoising functions In the above description of AMP, we assumed that the signal vector $\boldsymbol{\beta}_0$ had i.i.d. entries. This resulted in separable (scalar) denoising functions $\eta_t(\cdot)$. However, this assumption is prohibitive in compressed sensing applications where the signal vector $\boldsymbol{x}$

[6] The potential function can either be constructed explicitly to have the desired properties as done in [83], or it can be obtained by using the (non-rigorous) replica method in statistical physics as done in [56, 78].

represents real signals, e.g., image and audio signals. Simulation results in compressed imaging demonstrate that adapting the AMP algorithm's denoising function $\eta_t(\cdot)$ to incorporate more general signal models (e.g., Hidden Markov tree priors), or simply replacing them with popular image denoisers such as BM3D [88], greatly improves the quality of image reconstruction [89–92]. The corresponding state evolution recursions for these AMP algorithms have been theoretically justified for some classes of non-separable denoising functions [93–95]. When the signal distribution or certain parameters of the distribution are unknown, several methods have been proposed to incorporate *learning* or *universal denoising* into the AMP iteration [96–99].

Generalised linear models A generalised AMP (GAMP) algorithm and its corresponding state evolution and potential function has been proposed and analysed for generalised linear models, where the entries of the measurement vector $\boldsymbol{y}$ are obtained from $\boldsymbol{A}\boldsymbol{\beta}_0$ via a memoryless scalar channel [64,100,101]. These models are useful for modelling non-linearities at the output. For example, logistic regression corresponds to using the logistic function as the scalar channel and compressive phase retrieval [102] corresponds to using the absolute value function.

In addition to the LASSO, AMP (and GAMP) can also be used to analyse the asymptotic performance of other statistical estimation problems, including M-estimation [103], logistic regression [104, 105], and SLOPE [106].

Non-Gaussian matrices A crippling problem with the AMP algorithm is that it is sensitive to the choice of sensing matrix $\boldsymbol{A}$ in the linear (or generalised linear) model. The accuracy of the state evolution has only been proven for sensing matrices with i.i.d. Gaussian [76,100] and i.i.d. sub-Gaussian entries [107]. In numerical simulations, partial Fourier and Hadamard design matrices can be used, but state evolution recursions of the form given in (1.19) do not accurately track the performance of the AMP algorithm for such matrices. Furthermore, when the matrix $\boldsymbol{A}$ is ill-conditioned, the AMP (and GAMP) algorithm diverges. Hence, for problems where the matrix $\boldsymbol{A}$ is populated with data (e.g., generalised linear regression applications), there is no guarantee on the performance of the AMP or GAMP algorithm.

A few approaches have been considered to address this limitation. One approach is to design different algorithms using similar principles as that used to derive the AMP algorithm. For example, many algorithms have been obtained by using different approximations to loopy belief propagation and free energy/potential functions [108–111]. Although simulation results show that these algorithms are more robust to the choice of $\boldsymbol{A}$, their convergence properties have not yet been rigorously analysed. Another approach is to consider a larger class of random matrices [112, 113]. In [113], the vector AMP algorithm and its corresponding state evolution was proposed. It was proven that the proposed state evolution tracks the performance of VAMP for random matrices that are right-rotationally invariant.

Finally, one can view AMP and GAMP as optimisation algorithms and obtain convergence

guarantees for arbitrary $\boldsymbol{A}$ matrices. In [114], the authors provided convergence guarantees for AMP (GAMP) algorithms with added *damping* when the objective function is quadratic. The optimisation algorithm proposed in [115] is motivated by (and similar to) GAMP. It has convergence guarantees for arbitrary $\boldsymbol{A}$ matrices and strictly convex and smooth objective functions.

Bilinear models The AMP and GAMP algorithms have also been extended to (generalised) bilinear models for applications such as sparse PCA, low-rank matrix estimation, completion and factorisation, and dictionary learning [66, 116–122].

In this thesis we use the AMP algorithm as a decoder for SPARCs (introduced in Section 1.3) and their generalisations, and analyse the per-iteration performance of the AMP decoder via state evolution. In particular, we show that for any rate less than the capacity, state evolution analysis predicts that the asymptotic MSE (and section error rate) of the AMP decoder for spatially coupled SPARCs can be upper bounded by an arbitrarily small constant after a finite number of iterations (Chapter 2). In Chapter 3, we show that this result also extends to K-ary PSK modulated SPARCs (either power allocated or spatially coupled) when the modulation factor K is fixed. Furthermore, the results in [1, 22, 57] prove that the error performance of the AMP decoder we propose concentrates on their corresponding state evolution predictions. Based on the work on spatially coupled SPARCs, we also propose spatially coupled coding schemes for many-user Gaussian MACs (introduced in Section 1.2) and use the AMP algorithm for decoding (Chapter 4). We use state evolution and potential function analysis to show that this coding scheme can asymptotically achieve near-optimal trade-offs.

Compressed coding *Compressed coding* is a scheme that uses similar AMP techniques for communication over single-user and multiple-access Gaussian channels [123–126]. In single-user compressed coding, a coded modulation scheme (e.g., a binary code plus PSK) is first used to generate a code sequence $\boldsymbol{\beta}_0 \in \mathbb{R}^n$, which is then used to generate the channel input sequence $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\beta}_0$ via a random "compression" matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. An AMP algorithm is used to recover the code sequence $\boldsymbol{\beta}_0$ from the noisy channel output $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\beta}_0 + \boldsymbol{w}$. (The AMP iterates incorporate the demodulation/decoding procedures of the coded modulation scheme.) Using state evolution and the area property of extrinsic information transfer charts, [126] showed that the rate of compressed coding can asymptotically approach the channel capacity provided that either a certain curve matching condition is satisfied, or spatial coupling is used. Although state evolution is shown to track the proposed AMP decoder's error performance via simulations, a rigorous proof has not yet been shown.

1.5 Spatial coupling

1.5.1 Spatial coupling in LDPC codes

Spatial coupling is a technique originally developed for low density parity check (LDPC) codes to improve the threshold of the low complexity belief propagation (BP) decoder, which is known as the BP threshold.[7] The idea of spatial coupling emerged in the context of constructing convolutional codes using block codes [127], and in particular from LDPC codes [128]. Therefore, spatially coupled LDPC codes are also called LDPC convolutional codes.

The idea behind spatial coupling is to couple together several *regular* LDPC codes in a chain and use the same BP decoder as uncoupled regular LDPC codes. Since an LDPC code is defined by its parity check matrix which can be represented by a sparse graph, the coupling of LDPC codes can be represented by the coupling of disjoint sparse graphs; hence the name *spatial coupling*. See Figs. 1.8 and 1.9 for a graphical representation of the coupling procedure known as unwrapping [127, 128]. Many other coupling methods exist in the literature; in particular, the works of this thesis is inspired by protograph based spatially coupled LDPC codes [129,130].

Perhaps surprisingly, when the coupling procedure is properly terminated at the two ends of the chain, the BP threshold of the spatially coupled LDPC code improves over that of the underlying regular LDPC code. Furthermore, [14] proved that for the binary erasure channel (BEC), the threshold of a spatially coupled LDPC code under sub-optimal BP decoding matches the threshold of the underlying regular LDPC code under optimal maximum *a posteriori* (MAP) decoding — a phenomenon known as *threshold saturation*. Take the (3,6)-regular LDPC code for example, spatial coupling improves the threshold under BP decoding from $\epsilon_{\mathrm{BP}} \approx 0.4294$ to $\epsilon_{\mathrm{MAP}} \approx 0.4882$. The threshold saturation result for spatially coupled LDPC codes was later extended to general binary-input memoryless output-symmetric channels in [15]. Since the MAP decoding threshold of regular LDPC codes approaches the Shannon limit ($(1 - \text{rate})$ for the BEC) as the variable and check node degrees tend to infinity, spatially coupled LDPC codes with BP decoding is a practical coding scheme that is provably capacity achieving. A review on the theory and practice of spatially coupled LDPC codes is given in [131].

An intuitive explanation for the improvement in BP threshold is as follows. Consider the BEC for simplicity, noting that in this case BP decoding is equivalent to iteratively solving the parity check equations to recover the erased codeword bits/symbols. Notice in Fig. 1.9c that the check nodes at the two ends of the coupled graph are of a lower degree than the check nodes in the middle (which have the same degree as the underlying (3,6)-regular LDPC code). This allows the variables (codeword bits/symbols) at the two ends to be decoded more easily than the variables in the middle. Therefore, the variables at the two ends can start decoding at

[7] In coding theory, the "threshold" is the unique channel parameter that separates decoding success and failure. For example, consider a binary erasure channel with erasure probability ϵ and a (3,6)-regular LDPC code with BP threshold $\epsilon_{\mathrm{BP}} \approx 0.4294$. For $\epsilon < \epsilon_{\mathrm{BP}}$, BP decoding succeeds with high probability (for large code lengths), whereas for $\epsilon > \epsilon_{\mathrm{BP}}$, BP decoding fails with high probability.

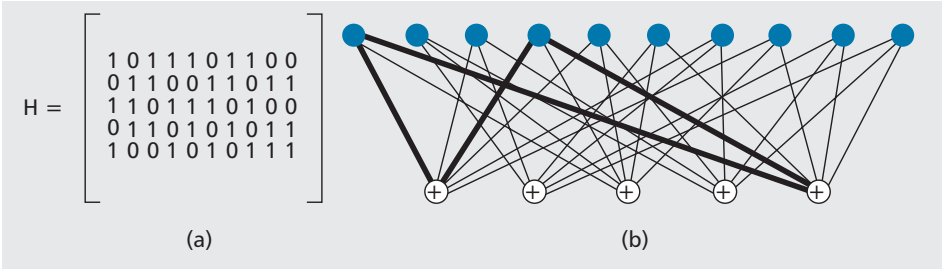


Figure 1.8: a) Parity-check matrix of a (3, 6)-regular LDPC block code with block length $n = 10$; b) the associated (3, 6)-regular Tanner graph. The blue circles represent code bits, or variable nodes; the open circles represent parity checks, or constraint nodes, and the darkened edges represent a cycle of length 4. Taken from Fig. 2 of [131], © 2014 IEEE.

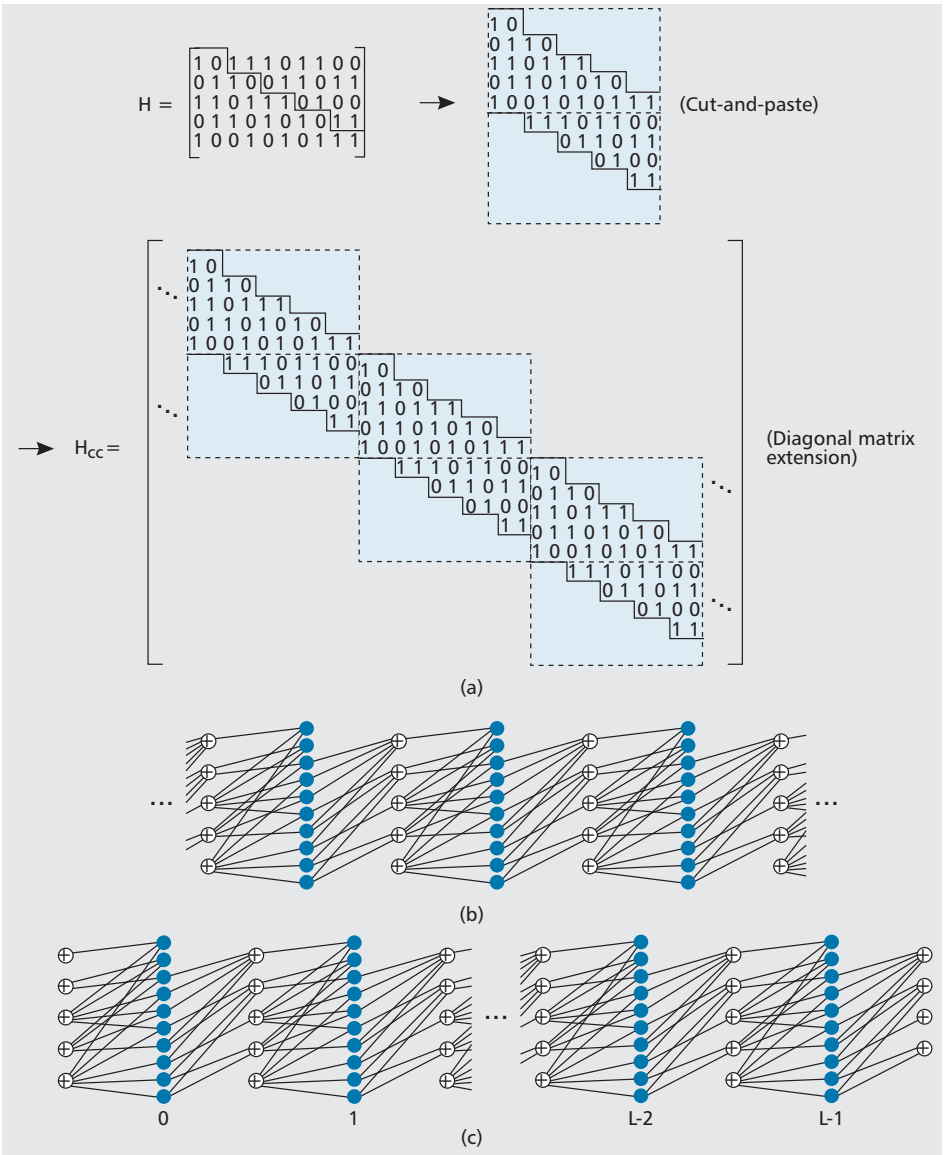


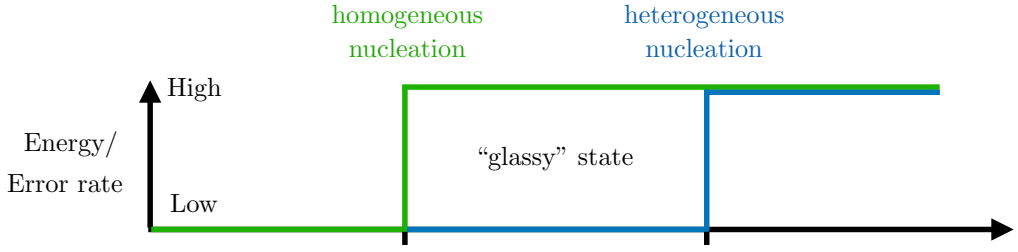
Figure 1.9: a) An illustration of the unwrapping procedure for a (3, 6)-regular LDPC block code; b) the Tanner graph associated with the unwrapped (3, 6)-regular LDPC convolutional code; c) the terminated Tanner graph associated with the unwrapped (3, 6)-regular LDPC convolutional code. Taken from Fig. 3 of [131], © 2014 IEEE.

erasure probabilities above the BP threshold of an uncoupled (3,6)-regular LDPC code. Once decoding is initiated at the two ends, a decoding "wave" propagates from the two ends until it reaches the middle.

Due to the extra check nodes at the two ends, spatial coupling does come at the cost of having a lower overall rate compared to the underlying uncoupled LDPC code. However, as the coupling length (the number of LDPC codes coupled together) increases, this rate loss becomes negligible.

Statistical physics analogy

The supercooling phenomenon — the cooling of a liquid below its freezing point without it becoming a solid — is a useful analogy in explaining the threshold saturation phenomenon in spatially coupled systems [65]. Under standard conditions, water freezes (crystallises) when the temperature is lowered to 0°C; however, if the water is pure and free of nucleation sites for crystallisation (e.g. after demineralisation), then water can be *supercooled* down to −48.3°C, which is when homogeneous nucleation occurs. In between 0°C and −48.3°C, pure water gets trapped in an amorphous (or "glassy") state. It is the impurities in the water acting as seed crystals that allow water to crystallise at 0°C. If a nucleation site is introduced to pure water in between 0°C and −48.3°C, then heterogeneous nucleation initiates from the boundary of the nucleation site and a nucleation "wave" propagates from the boundary to the rest of the liquid. A quick search for "supercooled water" on Youtube.com gives some good examples captured on video.



LDPC	ϵ_{BP}	ϵ_{MAP}	erasure probability
Compressed sensing	$\frac{1}{\delta_{\text{AMP}}(p_{\beta_0})}$	$\frac{1}{\bar{d}(p_{\beta_0})}$	compression ratio
SPARC	R_{AMP}	R_{MMSE}	rate

Figure 1.10: Toy figure illustrating the connections between the thresholds in various random systems. The LDPC thresholds are discussed on page 29, the (noiseless) compressed sensing thresholds in Section 1.5.2, and the SPARC thresholds in Section 1.5.3.

Regular LDPC codes can be likened pure water in the above example, and their BP threshold to $-48.3°$C. Since the parity check matrix is regular, the system is homogenous, without any "impurities" where decoding (crystallisation) can easily initiate. In spatially coupled LDPC codes, the extra check nodes at the two ends act as nucleation sites introduced to the system, from which decoding can initiate. The MAP threshold can be likened to $0°$C. Furthermore, the rest of the system away from the two ends (the "boundary") is just like a like a regular LDPC code without any "impurities". *Irregular* LDPC codes [132] can be seen as another way to introduce heterogeneity into the system. In Fig. 1.10, we draw a toy figure to illustrate the relationships between LDPC thresholds and water crystallisation temperatures.

We will repeatedly refer to this supercooling analogy when giving intuitive explanations to the workings of spatial coupling in other applications.

Threshold saturation proofs

Multiple strategies have been used to prove that the BP threshold of spatially coupled LDPC codes "saturate" the MAP threshold of the underlying regular LDPC code. The first proof was done via the Maxwell construction, which relates the fixed point of the coupled density evolution (DE) to the area threshold of the underlying regular LDPC code [14, 15, 133]. Later on, a strategy based on extrinsic information transfer (EXIT) charts [134] and strategies based on potential functions [83, 135] were used to provide a general proof of the threshold saturation phenomenon in a variety of applications, such as LDPC codes and compressed sensing. These proofs require the state of the underlying *uncoupled* system to be characterised by a scalar, e.g., the erasure probability characterises the state of density evolution.

The proof based on potential functions is the simplest of the three. The proof first constructs a potential function whose stationary points have a one-to-one correspondence with the fixed points of the scalar recursion corresponding to the uncoupled system. For example, in compressed sensing with AMP signal recovery (introduced in Section 1.4), the potential function in (1.25) has stationary points which correspond to the fixed points of state evolution (1.19). It is known that the (global) minimum of such a potential function is related to the optimal performance of the underlying system, e.g., MAP or MMSE decoding performance. Threshold saturation is then proved by showing that the coupled recursion corresponding to the spatially coupled system (e.g. coupled DE) has a unique fixed point that corresponds to the minimum of the potential function.

There are deep and fascinating connections between the fundamental properties of random systems, potential (free energy) functions, and message passing algorithms due to their connections with statistical physics. A recent survey article [136] reviews of some recent developments in the use of statistical physics ideas in statistical inference problems. Please refer to the references in [136] for detailed discussions on these topics.

In the following two sections, we discuss how spatial coupling has been applied to compressed

sensing and sparse regression codes.

1.5.2 Spatial coupling in compressed sensing

Recall from Section 1.4 that in compressed sensing one aims to recover a (possibly sparse) signal $\boldsymbol{\beta}_0 \in \mathbb{R}^n$ from noisy random linear measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\beta}_0 + \boldsymbol{w}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the sensing matrix and the noise vector $\boldsymbol{w} \in \mathbb{R}^m$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. The measurement ratio is given by $\delta = m/n$. In the noiseless case ($\sigma^2 = 0$), it is of great interest to find the minimum δ (maximum compression ratio) such that signal recovery is possible.

Consider compressed sensing in the setting where the sensing matrix $\boldsymbol{A}$ has i.i.d. Gaussian entries and the signal vector $\boldsymbol{\beta}_0$ has i.i.d. entries drawn from p_{β_0}. In this setting, when the measurements are noiseless ($\sigma^2 = 0$), signal recovery is lossless if and only if $\delta > \overline{d}(p_{\beta_0})$, where $\overline{d}(p_{\beta_0})$ is the upper Rényi information dimension of the signal [137]. One can compare this with the $\epsilon < \epsilon_{\text{MAP}}$ condition for successful MAP decoding of regular LDPC codes over the BEC.

However, when we consider the low complexity AMP algorithm described in Section 1.4 (using the MMSE estimator (1.21) as the denoising function), signal recovery is lossless if and only if $\delta > \delta_{\text{AMP}}(p_{\beta_0})$, where the AMP threshold $\delta_{\text{AMP}}(p_{\beta_0})$ is greater than or equal to $\overline{d}(p_{\beta_0})$. One can compare the AMP threshold to the BP threshold of regular LDPC codes. Take p_{β_0} to be the Bernoulli-Gaussian signal distribution for example, where each entry of $\boldsymbol{\beta}_0$ is drawn from a standard Gaussian with probability α, and set to zero with probability $1 - \alpha$. If we take $\alpha = 0.2$, then $\overline{d}(p_{\beta_0}) = 0.2$ and $\delta_{\text{AMP}}(p_{\beta_0}) \approx 0.355$. Note that $\overline{d}(p_{\beta_0})$ always equals the sparsity ratio α for Bernoulli-Gaussian priors.[8] See Fig. 1.10 for a toy figure illustrating the connections between thresholds in compressed sensing and LDPC codes.

The idea of using spatial coupling in compressed sensing was first proposed in [138]. Then in [65] it was demonstrated via numerical simulations that spatial coupling can improve the AMP threshold to near-optimal thresholds. A threshold saturation result similar to that in spatially coupled LDPC codes was established in [139], and also included in the threshold saturation proofs for general coupled systems [83, 134]. Specifically, [139] proved that lossless signal recovery is achieved with spatial coupling and AMP if the measurement ratio δ exceeds the theoretically optimal threshold $\overline{d}(p_{\beta_0})$. Furthermore, [139] proved that such a signal recovery scheme is robust to measurement noise ($\sigma^2 > 0$).

In LDPC codes, spatial coupling involves the coupling of *sparse* factor graphs; in compressed sensing, spatial coupling involves the coupling of *dense* factor graphs corresponding to the dense sensing matrix $\boldsymbol{A}$. Each factor graph has n variable nodes and m constraint nodes corresponding to the signal entries and measurement constraints, respectively (see [77] for more details). The resulting spatially coupled system corresponds to using a block-diagonal sensing matrix $\boldsymbol{A}$ in

[8] This is very surprising, it tells us that the optimal signal recovery scheme can recover the signal as if knowing the exact locations of the non-zero entries. More specifically, the number of "extra" measurements optimal scheme requires is sublinear in n.

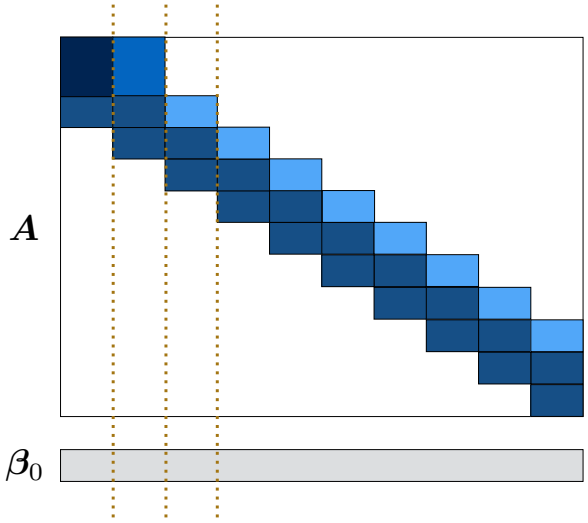


Figure 1.11: Spatially coupled sensing matrix $\boldsymbol{A}$. Entries within the same blue block are i.i.d. zero mean Gaussian, and entries outside the block-diagonal region (the white areas) are equal to zero. Different shades of blue represent different Gaussian variances. Dashed lines are drawn to show how blocks in the sensing matrix correspond to blocks in the signal vector. (Inspired by [56, Fig. 5].)

compressed sensing. An example spatially coupled sensing matrix is illustrated in Fig. 1.11, where the entries within the same blue block are i.i.d. Gaussian, and entries outside the block-diagonal region (the white areas) are equal to zero. Entries across blocks are independent and can have different Gaussian variances, as illustrated by the different shades of blue.

Each block of the spatially coupled sensing matrix in Fig. 1.11 can be seen as an uncoupled sensing matrix. The signal vector $\boldsymbol{\beta}_0$ and noisy measurement vector $\boldsymbol{y}$ can be partitioned into blocks that correspond to the *column-blocks* and *row-blocks* of the spatially coupled sensing matrix, respectively. Consider the second row-block of the sensing matrix in Fig. 1.11, which has three non-zero blocks. This means that the second row-block of $\boldsymbol{y}$ contains information about the first three column-blocks of $\boldsymbol{\beta}_0$. Although many different spatially coupled sensing matrix constructions have been proposed in the literature [56,78,124,139,140], they are all have a block-diagonal structure, and many of them have an explicit "seed" region at the top-left corner of the sensing matrix. In Fig. 1.11, this "seed" region is constructed by adding more rows (measurements) to the blocks in the first row-block, which results in those blocks being "taller" (having higher measurement ratio) compared to other blocks. The "seed" region helps jump start AMP decoding, just like the extra check nodes at the two ends of spatially coupled LDPC codes (see Fig. 1.9c) help jump start BP decoding.

1.5.3 Spatial coupling in sparse regression codes

Since sparse regression codes (SPARCs) can be seen as a special case of compressed sensing where the signal vector $\boldsymbol{\beta}_0$ has a *section-wise* i.i.d. structure (compare the beginnings of Sections 1.3 and 1.4), spatial coupling ideas and AMP decoding algorithms have been applied to SPARCs.

AMP decoding algorithms were first applied to SPARCs in [54], and similar authors introduced spatially coupled SPARCs with AMP decoding [55, 56]. The authors proposed spatially coupled design matrices for SPARCs that are of the type shown in Fig. 1.11. The analysis of spatially coupled SPARCs under AMP decoding in these works were based on state evolution and heuristic statistical physics methods such as the replica method. Their (not fully rigorous) analysis showed that spatially coupled SPARCs with "flat" power allocation and AMP decoding are asymptotically capacity achieving. Their numerical simulation results showed that the error rate of spatially coupled SPARCs drops faster than that of power allocated SPARCs as the code rate moves away from the channel capacity. Later on, the same authors made some steps of the analysis rigorous [84, 140]. In particular, they proved a threshold saturation result similar to that for spatially coupled LDPC codes and compressed sensing, and also showed that the optimal (rate) threshold R_{MMSE} of uncoupled SPARCs approaches the channel capacity as the section size parameter M tends to infinity.[9] The latter result is similar to how the MAP threshold of regular LDPC codes ϵ_{MAP} tends to the Shannon limit as the degrees of the code tend to infinity. See Fig. 1.10 for a toy figure illustrating the connections between thresholds in LDPC codes, compressed sensing and SPARCs.[10]

In this thesis we also consider spatially coupled SPARCs (Chapter 2). We propose a simpler construction for spatially coupled design matrices compared to earlier works, and show that this construction provides a unified framework to analyse both power allocated and spatially coupled SPARCs. It even enables one to consider SPARCs that utilise both techniques simultaneously. Using the simpler spatially coupled construction, we provide an alternative proof to threshold saturation to show that spatially coupled SPARCs with AMP decoding are asymptotically capacity achieving. We also provide numerical simulation results to show the finite code length performance of spatially coupled SPARCs in comparison to power allocated SPARCs and standard coded modulation schemes.

In contrast to threshold saturation, which analyses the fundamental threshold of the system and the fixed points of a coupled state evolution, we analyse the iteration-by-iteration performance of the AMP decoder via a coupled state evolution. This can be likened to the *static* versus *dynamic* analysis in physics, where the former analyses the equilibrium state of a system (i.e., when the time index tends to infinity and the system settles down), and the latter analyses how the state of a system evolves over time. From another point of view, our analysis involves taking limits in a different order compared to threshold saturation. In threshold saturation, one first takes the coupling length (the number of graphs coupled together) to infinity and then a certain code parameter to infinity (for SPARCs it is the section size M). In our analysis we first consider a sufficiently large section size M and then consider a sufficiently large coupling

[9] In the context of SPARCs, the threshold is the parameter such that for rates less than the threshold, decoding succeeds with high probability, and for rates larger than the threshold, decoding fails with high probability.

[10] In Chapter 2, R_{AMP} is written as R_{BP} to follow the convention in the literature [56].

length. The benefit of our analysis is that we can understand the (large M) performance of the AMP decoder at any iteration. This allows us to understand the decoding progression in spatially coupled SPARCs and provide an upper bound on the number of iterations required for successful decoding. By understanding the wave-like decoding progression of spatially coupled SPARCs, we introduce a *sliding window* AMP decoder which has a per-iteration complexity that is independent of the code length.

In Chapter 4, we apply the same techniques to construct and analyse spatially coupled coding schemes with AMP decoding for many-user Gaussian MACs. We obtain a threshold saturation result in terms of the user error rate (fraction of user messages decoded in error) instead of the conventional MSE, and show that our proposed coding scheme can achieve near-optimal user density and signal-to-noise ratio trade-offs.

1.5.4 Other applications

Spatial coupling has been successfully applied in many applications. A non-exhaustive list includes random constraint satisfaction problems in computer science [141, 142], lossy compression [143], multiple access [144–147], and the Curie-Weiss model in statistical physics [148, 149]. Due to the generality of the threshold saturation phenomenon, spatial coupling has also been used as a proof technique to study difficult theoretical problems [84, 150]. In these works, one would analyse the optimal threshold of a system by analysing the fixed point of a hypothetically constructed coupled system. An overview of the applications of spatial coupling is given in [15, Sec. I].

1.6 Structure of thesis

- In Chapter 2 we introduce and analyse spatially coupled (SC) SPARCs for communication over the (single-user) AWGN channel. In Section 2.1, we introduce base matrices for the construction of SPARC design matrices, which provide a unified framework for designing power allocated and spatially coupled SPARCs. In Section 2.2 we introduce the AMP decoder and the state evolution recursion that tracks the per-iteration error of the AMP decoder.

  In Section 2.3, we analyse the decoding progression of the AMP decoder for a simple SC-SPARC construction that is defined by the coupling width ω and coupling length Λ of the coupled system. In particular, for the simple coupling structure and for large message vector section sizes, we obtain an upper bound on a key state evolution parameter that tracks the per-iteration mean squared error (MSE) of the AMP decoder, which translates to an upper bound on the per-iteration section error rate (SER). The analysis helps explain the wave-like decoding progression in SC-SPARCs, and shows that for any rate less than the capacity, and sufficiently large coupling width ω and coupling length Λ,

the AMP decoder achieves perfect decoding (zero SER as the code length $n \to \infty$) in a finite number of iterations which is inversely proportional to the rate gap to capacity. In Section 2.4, we use the above result together with the recent AMP concentration result in [1, Thm. 2] to show that the probability of excess section error rate decays exponentially in the code length n.

In Section 2.5, we provide finite code length simulation results of SC-SPARCs with AMP decoding and compare them with power allocated SPARCs and standard coded modulation schemes. In Section 2.6, we introduce a sliding window AMP decoder that takes advantage of the wave-like decoding progression of SC-SPARCs, whose decoding latency and per-iteration computational complexity is independent of the code length.

- In Chapter 3, we introduce and analyse modulated (complex) SPARCs for communication over the complex AWGN channel. In a modulated SPARC, information is encoded in both the locations and the values of the non-zero entries of the message vector, with the value of each non-zero entry chosen from a set of K values. When $K = 1$, we recover regular SPARCs without modulation (introduced in Section 1.3).

  In Section 3.2, we introduce modulated SPARCs with K-ary phase shift keying (PSK) modulation and justify our choice of using PSK. We consider modulated SPARCs whose design matrix is constructed using a base matrix (as in Chapter 2). In Section 3.3, we introduce the AMP decoder for K-PSK modulated (complex) SPARCs and the state evolution recursion which tracks the per-iteration error of the AMP decoder.

  In Section 3.4, we analyse the error performance of the AMP decoder for modulated SPARCs using state evolution. The main technical result (Proposition 3.4.1) gives an upper bound on a key state evolution parameter which tracks the per-iteration MSE of the AMP decoder, which translates to an upper bound on the per-iteration SER. Using this bound, we show that in limit of large message vector section size, the state evolution of K-PSK modulated SPARCs is the same for any fixed value of K, including $K = 1$ (unmodulated). We use this result to prove that K-PSK modulated SPARCs with AMP decoding are asymptotically capacity achieving for the complex AWGN channel, with either spatial coupling or exponentially decaying power allocation (Theorems 2 and 3).

  In Section 3.5, we provide finite code length simulation results of modulated SPARCs with AMP decoding and compare them with coded modulation schemes using LDPC codes from the DVB-S2 standard [151]. The results demonstrate that using modulation in SPARCs can significantly reduce the decoder complexity without sacrificing error performance.

- In Chapter 4 we study the many-user Gaussian multiple access channel in the asymptotic regime where the number of users L grows linearly with the code length n, i.e., both L and n tend to infinity with user density $\mu = L/n$ held constant. We are interested in

finding the optimal asymptotic trade-off between the signal-to-noise ratio E_b/N_0 and the user density μ, for a fixed target error rate and number of bits to be transmitted per user (user payload).

In Section 4.2, we introduce coding/multiple-access schemes base on random linear models with AMP decoding. We derive the exact asymptotic user error rate (the fraction of user messages decoded in error) achieved by these schemes using state evolution and potential function analysis, and find that the asymptotic achievability of a coding scheme based on spatially coupled Gaussian matrices and AMP decoding nearly matches the converse bound for a large range of user densities. The spatially coupled scheme can be interpreted as generalised time-sharing: the coupling structure specifies which users are active during each channel use.

In Section 4.3, we analyse the performance of these coding schemes as the user payload grows large and discuss how using smaller codebooks and modulation can help avoid the high complexity at large user payloads.

In Section 4.4, we extend our results to complex random linear coding schemes for the complex Gaussian multiple access channel in the many-user setting.

- In Chapter 5, we summarise the work of this thesis and propose potential directions for future research.

1.7 Notation

Table 1.1 on page 39 shows a list of the notations and acronyms used in this thesis.

Table 1.1: Notations and acronyms used in this thesis.

Notation/acronym	Description
i.i.d.	independent and identically distributed
w.r.t.	with respect to
a.s.	almost surely
LHS, RHS	left-hand-side, right-hand-side
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
a	Scalars are usually denoted by lower case English or Greek letters
$\boldsymbol{a}$	Vectors are usually denoted by bold lower case English or Greek letters
$\boldsymbol{A}$	Matrices are usually denoted by bold upper case English or Greek letters
a_i	The i-th entry of vector $\boldsymbol{a}$
$A_{i,j}$	The entry at the i-th row and j-th column of matrix $\boldsymbol{A}$
$\boldsymbol{A}^*$	The conjugate transpose of matrix $\boldsymbol{A}$
$\Re(z)$	The real part of a complex number z. If $z = x + \mathrm{j}\, y$, then $\Re(z) = x$.
$\Im(z)$	The real part of a complex number z. If $z = x + \mathrm{j}\, y$, then $\Im(z) = y$.
$\overline{z}$	The complex conjugate of a complex number z.
$\boldsymbol{I}_N$	The $N \times N$ identity matrix
$\|\boldsymbol{a}\|^2$	The squared ℓ_2-norm of vector $\boldsymbol{a}$, i.e., $\|\boldsymbol{a}\|^2 \coloneqq \sum_i \|a_i\|^2$.
ln, $\log_2$	Natural logarithm, base 2 logarithm, and logarithm with undefined
log	Logarithm with undefined base (either unnecessary or clear from context)
$\mathbb{1}\{\mathcal{A}\}$	Indicator function of an event $\mathcal{A}$
$[N]$	The set $\{1, \ldots, N\}$ for a positive integer N
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$\mathcal{CN}(\mu, \sigma^2)$	Circularly symmetric complex Gaussian distribution with mean μ and variance σ^2
$O(x_n)$, $o(x_n)$	For deterministic sequences $(s_n)_{n\geq 0}, (x_n)_{n\geq 0}$, we write $s_n = O(x_n)$ if $\frac{\|x_n\|}{s_n}$ is bounded above by a strictly positive constant for all sufficiently large n, and $s_n = o(x_n)$ if $\lim_{n\to\infty} \frac{\|x_n\|}{s_n} = 0$.

Chapter 2

Spatially coupled sparse regression codes

In this chapter we introduce spatially coupled sparse regression codes (SPARCs) for communication over the (single-user) AWGN channel. We introduce *base matrices* for the construction of SPARC design matrices, which provides a unified framework for designing power allocated and spatially coupled SPARCs (Section 2.1). We also introduce the approximate message passing (AMP) decoder and the corresponding state evolution recursion which tracks the per-iteration asymptotic error of the AMP decoder (Section 2.2).

In the limit of large section size M, we obtain a succinct characterisation of the state evolution recursion. We analyse this *asymptotic state evolution* for a simple spatial coupling construction that is defined by the coupling width ω and coupling length Λ. The analysis shows that for any rate less than the capacity, and sufficiently large coupling width ω and coupling length Λ, the AMP decoder achieves perfect decoding (zero section error rate as the code length $n \to \infty$) in a finite number of iterations which is inversely proportional to the rate gap to capacity (Section 2.3). The analysis also explains the "wave-like" decoding progression in spatially coupled SPARCs. The asymptotic state evolution result is then refined for large but finite section sizes M, which we use together with the recent AMP concentration result in [1, Thm. 2] to show that the probability of excess section error rate decays exponentially in the code length n (Section 2.4).

In Section 2.5, we provide finite code length simulation results of spatially coupled SPARCs with AMP decoding and compare them with power allocated SPARCs and standard coded modulation schemes. Motivated by the wave-like decoding progression of spatially coupled SPARCs, we introduce a *sliding window* AMP decoder in Section 2.6 whose per-iteration complexity is independent of the code length.

In this chapter the rate and capacity terms (e.g., R and C) are given in *bits* for the discussion of simulation results (Sections 2.5 and 2.6) and in *nats* elsewhere.

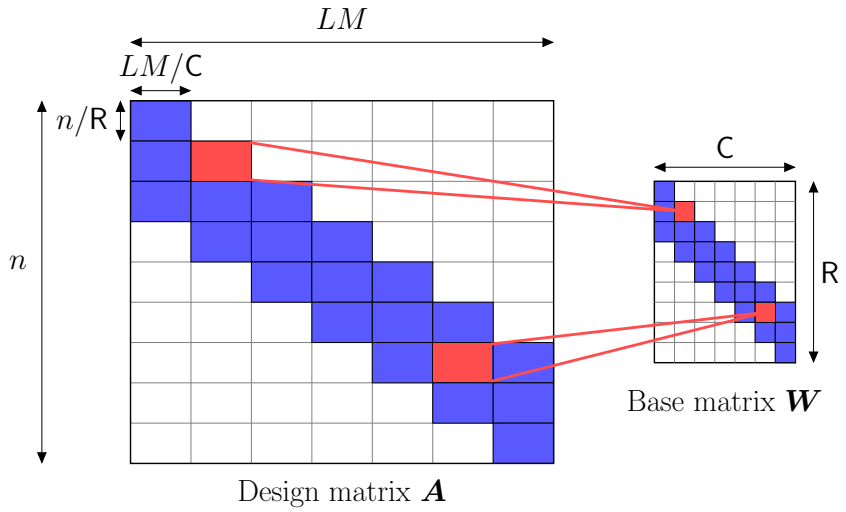


Figure 2.1: A spatially coupled design matrix $\boldsymbol{A}$ is divided into blocks of size $\frac{n}{\mathsf{R}} \times \frac{LM}{\mathsf{C}}$. There are R and C blocks in each column and row respectively. The entries in each block of $\boldsymbol{A}$ are i.i.d. Gaussian with zero mean and variance specified by the corresponding entry of the base matrix $\boldsymbol{W}$. Each square in $\boldsymbol{W}$ represents a scalar entry which specifies the variance of the entries in a block of $\boldsymbol{A}$. The base matrix shown here is an (ω, Λ, ρ) base matrix (Def. 2.1.1) with parameters $\omega = 3$, $\Lambda = 7$ and $\rho = 0$. The white parts of $\boldsymbol{A}$ and $\boldsymbol{W}$ correspond to zeros.

2.1 Spatially coupled SPARC construction

As in the standard construction of SPARCs (see Section 1.3), a spatially coupled (SC) SPARC is defined by a design matrix $\boldsymbol{A}$ of dimension $n \times LM$, where n is the code length. The codeword $\boldsymbol{x} \in \mathbb{R}^n$ is generated by $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\beta}$, where the message vector $\boldsymbol{\beta}$ has one non-zero entry in each of its L sections (M entries in each section). Without loss of generality, we set the values of the non-zero entries in $\boldsymbol{\beta}$ to 1. (We will discuss later how to incorporate power allocation into our framework.)

In an SC-SPARC, the design matrix $\boldsymbol{A}$ consists of independent zero-mean Gaussian entries whose variances are specified by a *base matrix* $\boldsymbol{W}$ of dimension $\mathsf{R} \times \mathsf{C}$. The design matrix is obtained from the base matrix $\boldsymbol{W}$ by replacing each entry W_{rc} by an $(n/\mathsf{R}) \times (LM/\mathsf{C})$ block with i.i.d. entries $\sim \mathcal{N}(0, W_{\mathsf{rc}}/L)$, for $\mathsf{r} \in [\mathsf{R}]$, $\mathsf{c} \in [\mathsf{C}]$. This is analogous to the "graph lifting" procedure for constructing spatially coupled LDPC codes from protographs [130]. See Fig. 2.1 for an example.

From the construction, the design matrix $\boldsymbol{A}$ has independent Gaussian entries

$$A_{ij} \sim \mathcal{N}\Big(0, \frac{1}{L} W_{\mathsf{r}(i)\mathsf{c}(j)}\Big), \quad \text{for } i \in [n],\ j \in [LM]. \tag{2.1}$$

The operators $\mathsf{r}(\cdot) : [n] \to [\mathsf{R}]$ and $\mathsf{c}(\cdot) : [LM] \to [\mathsf{C}]$ in (2.1) map a particular row or column index in $\boldsymbol{A}$ to its corresponding *row-block* or *column-block* index in $\boldsymbol{W}$. We require C to divide L, resulting in L/C sections per column block.

Recall that the non-zero entries in $\boldsymbol{\beta}$ are all set to 1. Then in order to satisfy the average

power constraint $\|\boldsymbol{x}\|^2 = nP$ (in expectation), the entries of the base matrix $\boldsymbol{W}$ must satisfy

$$\frac{1}{\mathsf{RC}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}} = P. \tag{2.2}$$

The trivial base matrix with $\mathsf{R} = \mathsf{C} = 1$ corresponds to an uncoupled SPARC with "flat" power allocation, while a base matrix consisting of a single row $\mathsf{R} = 1$, $\mathsf{C} = L$ is equivalent to an uncoupled SPARC with power allocation. (The power allocation is in the base matrix entries instead of the values of the non-zero entries of $\boldsymbol{\beta}$.) For example, the exponentially decaying power allocation in (1.12) is equivalent to choosing the base matrix values as follows:

$$W_{1,\ell} = LP \cdot \frac{2^{2C/L} - 1}{1 - 2^{-2C}} \cdot 2^{-2C\ell/L} \quad \text{for } \ell \in [L], \tag{2.3}$$

where C is the capacity of the AWGN channel (in bits) given in (1.5).

We mention that the idea of constructing a power allocated or spatially coupled design matrix from a base matrix is inspired by the construction of a irregular or spatially coupled LDPC code from a protograph [129,130,132,152]. In the rest of this chapter, we use the following base matrix inspired by the coupling structure of spatially coupled LDPC codes constructed from protographs [130].

Definition 2.1.1. An (ω, Λ, ρ) base matrix $\boldsymbol{W}$ is described by three parameters: coupling width $\omega \geq 1$ coupling length $\Lambda \geq 2\omega - 1$, and $\rho \in [0, 1)$ which determines the fraction of power allocated to the coupled entries in each column. The matrix has $\mathsf{R} = \Lambda + \omega - 1$ rows, $\mathsf{C} = \Lambda$ columns, with each column having ω identical non-zero entries. For an average power constraint P, the (r, c)-th entry of the base matrix, for $\mathsf{r} \in [\mathsf{R}], \mathsf{c} \in [\mathsf{C}]$, is given by

$$W_{\mathsf{rc}} = \begin{cases} (1-\rho)P \cdot \frac{\Lambda+\omega-1}{\omega} & \text{if } \mathsf{c} \leq \mathsf{r} \leq \mathsf{c} + \omega - 1, \\ \rho P \cdot \frac{\Lambda+\omega-1}{\Lambda-1} & \text{otherwise.} \end{cases} \tag{2.4}$$

It is easy to verify that this definition satisfies the power constraint in (2.2). For example, the base matrix in Fig. 2.1 has parameters $\omega = 3$, $\Lambda = 7$ and $\rho = 0$. For our simulations in Section 2.5, we use $\rho = 0$, whereas for our theoretical result (Theorem 1) we choose ρ to be a small positive value proportional to the rate gap to capacity. (Choosing $\rho = 0$ causes some technical difficulties in the proof, which can be addressed by picking a suitable $\rho > 0$.) Other ways to construct spatially coupled design matrices can be found in [56, 78, 139, 140].

Each non-zero entry in a base matrix $\boldsymbol{W}$ corresponds to an $(n/\mathsf{R}) \times (ML/\mathsf{C})$ block in the design matrix $\boldsymbol{A}$. Each block can be viewed as an uncoupled SPARC with L/C sections (with M columns in each section), code length n/R, and rate $R_{\text{inner}} = \frac{(L/\mathsf{C}) \ln M}{(n/\mathsf{R})}$ nats. R_{inner} is related

to the the overall rate of the SC-SPARC $R = \frac{L \ln M}{n}$ according to

$$R_{\text{inner}} = \frac{\mathsf{R}}{\mathsf{C}} R = \left(1 + \frac{\omega - 1}{\Lambda}\right) R, \tag{2.5}$$

where the last equality holds for an (ω, Λ, ρ) base matrix.

With spatial coupling, the coupling width ω is an integer greater than 1, so $R < R_{\text{inner}}$. The difference $(R_{\text{inner}} - R)$ is sometimes referred to as the rate loss due to spatial coupling. From (2.5), we see that rate loss depends on the ratio $(\omega - 1)/\Lambda$, which becomes negligible when Λ is large with respect to (w.r.t.) ω. For our theoretical results, we will be interested in the regime where $L \gg \mathsf{C} = \Lambda \gg \omega$.

Remark 2.1.1. SC-SPARC constructions usually have a "seed" to jumpstart decoding. In [140], a small fraction of $\boldsymbol{\beta}$'s sections are fixed a priori — this pinning condition is used to analyse the state evolution equations via the potential function method. Analogously, the construction in [56] introduces additional rows in the design matrix for blocks corresponding to the first row of the base matrix. In an (ω, Λ, ρ) base matrix, the fact that the number of rows in the base matrix exceeds the number of columns by $(\omega - 1)$ helps decoding start from both ends. In Section 2.3.1, we describe how AMP decoding progresses in an $(\omega, \Lambda, \rho = 0)$ base matrix.

Remark 2.1.2. The base matrix framework described above allows one to design SPARCs with both power allocation and spatial coupling. To do this, first construct a spatially coupled (ω, Λ, ρ) base matrix $\boldsymbol{W}$. (Recall that each entry of the base matrix corresponds to an $(n/\mathsf{R}) \times (LM/\mathsf{C})$ block in the spatially coupled design matrix, which can be seen as an uncoupled design matrix with L/C sections.) Then, replace each entry of the base matrix with an $1 \times (L/\mathsf{C})$ vector. These vectors are obtained by multiplying the original value of the base matrix entry (possibly zero) with a $1 \times (L/\mathsf{C})$ power allocation vector (entry-wise). For example, the power allocation vector could be the exponentially decaying power allocation in (2.3) or the iterative power allocation scheme in [47] (with the number of sections equal to L/C instead of L). The resulting base matrix has dimensions $(\Lambda + \omega - 1) \times L$. In order for a SPARC with both power allocation and spatial coupling to have good error performance, both the spatial coupling parameters (coupling width and length) and the power allocation parameters (see [47]) need to be optimised. Preliminary simulation results show that it is possible for SPARCs with both power allocation and spatial coupling to achieve lower error rates than SPARCs with either power allocation or spatial coupling, at rates close to the channel capacity.[1] We leave a thorough study of such SPARC designs for future work.

In the remainder of the thesis, we use subscripts in sans-serif font (r or c) to denote row

[1] SPARCs with both power allocation and spatially coupled were considered in [56, Sec. VI] and the simulation results showed poor finite code length error performance. This is because the code was not properly designed and optimised.

or column block indices. Thus, $\boldsymbol{\beta}_{\mathsf{c}} \in \mathbb{R}^{LM/\mathsf{C}}$ denotes the c-th column block of $\boldsymbol{\beta} \in \mathbb{R}^{LM}$, for $\mathsf{c} \in [\mathsf{C}]$.

2.2 AMP decoder

Since the SC-SPARC codeword is $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\beta}$, the AWGN channel output $\boldsymbol{y} \in \mathbb{R}^n$ can be represented as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\beta} + \boldsymbol{w}, \tag{2.6}$$

where the noise vector $\boldsymbol{w} \in \mathbb{R}^n$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. The SC-SPARC decoder aims to recover the message vector $\boldsymbol{\beta}$ from the channel output $\boldsymbol{y}$. The design matrix $\boldsymbol{A}$ and the base matrix $\boldsymbol{W}$ are known to the decoder.

In this thesis we consider an approximate message passing (AMP) decoder for SC-SPARCs (see Section 1.4 for an introduction to AMP algorithms). The AMP decoder for SC-SPARCs can be derived using an approach similar to the one for uncoupled SPARCs [22, App. A], with modifications to account for the different variances for the blocks of $\boldsymbol{A}$ specified by the base matrix. It can also be derived from AMP algorithm for spatially coupled compressed sensing [139] after accounting for the section-wise prior of the message vector.

The AMP decoder iteratively generates message vector estimates $\boldsymbol{\beta}^t$ at iterations $t = 0, 1, 2, \ldots$ as follows: initialise $\boldsymbol{\beta}^0$ to the all-zero vector, and for $t \geq 0$ compute

$$\boldsymbol{z}^t = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}^t + \widetilde{\boldsymbol{v}}^t \odot \boldsymbol{z}^{t-1}, \tag{2.7}$$

$$\boldsymbol{\beta}^{t+1} = \eta^t(\boldsymbol{\beta}^t + (\widetilde{\boldsymbol{Q}}^t \odot \boldsymbol{A})^* \boldsymbol{z}^t). \tag{2.8}$$

Here $\odot$ denotes the Hadamard (entry-wise) product and $\boldsymbol{A}^*$ denotes the transpose of matrix $\boldsymbol{A}$. The vector $\widetilde{\boldsymbol{v}}^t \in \mathbb{R}^n$, the matrix $\widetilde{\boldsymbol{Q}}^t \in \mathbb{R}^{n \times ML}$, and the denoising function $\eta^t(\cdot)$ are defined below in terms of the state evolution parameters. Quantities with negative iteration indices are set to zero. After some rescaling and simplification, the AMP decoder proposed for SC-SPARCs in [56] is equivalent to the one shown above.

When the AMP decoder reaches the maximum allowed iteration number or an early stopping criterion (described in a later section), we take the latest estimate of $\boldsymbol{\beta}$, and set the largest entry in each section to 1 and the remaining entries to 0 to obtain the decoded message vector $\hat{\boldsymbol{\beta}}$.

State evolution Given a base matrix $\boldsymbol{W}$, state evolution (SE) is a deterministic recursion that iteratively defines a sequence of scalars $(\phi_{\mathsf{r}}^t)_{\mathsf{r} \in [\mathsf{R}]}$ and $(\psi_{\mathsf{c}}^t)_{\mathsf{c} \in [\mathsf{C}]}$, for $t \geq 0$. Initialise $\psi_{\mathsf{c}}^0 = 1$ for $\mathsf{c} \in [\mathsf{C}]$, and for $t = 0, 1, \ldots$, compute

$$\gamma_{\mathsf{r}}^t = \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}} \psi_{\mathsf{c}}^t, \qquad \phi_{\mathsf{r}}^t = \sigma^2 + \gamma_{\mathsf{r}}^t, \qquad \mathsf{r} \in [\mathsf{R}], \tag{2.9}$$

$$\tau_{\mathsf{c}}^t = \frac{R}{\ln M}\left[\frac{1}{\mathsf{R}}\sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\phi_{\mathsf{r}}^t}\right]^{-1}, \qquad \psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(\tau_{\mathsf{c}}^t), \qquad \mathsf{c} \in [\mathsf{C}], \tag{2.10}$$

where $\mathcal{E}(\tau_{\mathsf{c}}^t)$ is defined with $U_1, \ldots, U_M \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ as

$$\mathcal{E}(\tau_{\mathsf{c}}^t) = \mathbb{E}\left[\frac{e^{U_1/\sqrt{\tau_{\mathsf{c}}^t}}}{e^{U_1/\sqrt{\tau_{\mathsf{c}}^t}} + e^{-1/\tau_{\mathsf{c}}^t}\sum_{j=2}^M e^{U_j/\sqrt{\tau_{\mathsf{c}}^t}}}\right]. \tag{2.11}$$

SE parameters in the AMP decoder For $t \geq 1$, the vector $\widetilde{\boldsymbol{v}}^t \in \mathbb{R}^n$ in (2.7) and the matrix $\widetilde{\boldsymbol{Q}}^t \in \mathbb{R}^{n \times LM}$ in (2.8) have a block-wise structure (indicated by the tilde), with entries defined as follows. For $i \in [n], j \in [LM]$,

$$\widetilde{v}_i^t = \frac{\gamma_{\mathsf{r}(i)}^t}{\phi_{\mathsf{r}(i)}^{t-1}}, \qquad \widetilde{Q}_{ij}^t = \frac{\tau_{\mathsf{c}(j)}^t}{\phi_{\mathsf{r}(i)}^t}, \tag{2.12}$$

where we recall that $\mathsf{r}(i)$ and $\mathsf{c}(j)$ denote the row block index of the i-th entry and the column block index of the j-th entry, respectively. The vector $\widetilde{\boldsymbol{v}}^0$ is defined to be all-zeros.

Using (2.12) in (2.8), one can also write the update equation for $\boldsymbol{\beta}^{t+1}$ in the AMP decoder as follows,

$$\boldsymbol{\beta}^{t+1} = \eta^t\Big(\boldsymbol{\beta}^t + \widetilde{\boldsymbol{\tau}}^t \odot \Big[\boldsymbol{A}^*\Big(\big(\widetilde{\boldsymbol{\phi}}^t\big)^{-1} \odot \boldsymbol{z}^t\Big)\Big]\Big), \tag{2.13}$$

where the vectors $\widetilde{\tau}^t \in \mathbb{R}^{LM}$ and $\widetilde{\phi}^t \in \mathbb{R}^n$ have a block-wise structure, with entries $\widetilde{\tau}_j^t = \tau_{\mathsf{c}(j)}^t$ and $\widetilde{\phi}_i^t = \phi_{\mathsf{r}(i)}^t$ for $i \in [n], j \in [LM]$. The vector $(\widetilde{\boldsymbol{\phi}}^t)^{-1}$ denotes the element-wise inverse of $\widetilde{\boldsymbol{\phi}}^t$. This version of the update equation is used in implementation of the AMP decoder due to its lower computational complexity compared to (2.8).

The denoising function $\eta^t = (\eta_1^t, \ldots, \eta_{LM}^t) : \mathbb{R}^{LM} \to \mathbb{R}^{LM}$ in (2.8) is defined as follows, for $j \in [LM]$. For index j in section $\ell \in [L]$, with section ℓ in column block $\mathsf{c} \in [\mathsf{C}]$,

$$\eta_j^t(\boldsymbol{s}) = \frac{e^{s_j/\tau_{\mathsf{c}}^t}}{\sum_{j' \in \mathrm{sec}(\ell)} e^{s_{j'}/\tau_{\mathsf{c}}^t}}, \tag{2.14}$$

where $\mathrm{sec}(\ell) := \{(\ell-1)M + 1, \ldots, \ell M\}$ refers to the set of indices in section ℓ. We note that $\eta_j^t(\boldsymbol{s})$ depends on all the components of $\boldsymbol{s}$ in the section containing j.

Interpretation of the AMP decoder and state evolution The input to $\eta^t(\cdot)$ in (2.8), denoted by $\boldsymbol{s}^t$, can be viewed as a noisy version of true message vector $\boldsymbol{\beta}$. Consider an index j in section $\ell \in [L]$ which belongs to column block $\mathsf{c} \in [\mathsf{C}]$. Recall that $\boldsymbol{\beta}_\ell \in \mathbb{R}^M$ is section ℓ of the message vector, and let $\boldsymbol{s}_\ell^t$ denote section ℓ of the input vector to $\eta^t(\cdot)$. Then, $\boldsymbol{s}_\ell^t$ is approximately distributed as $\boldsymbol{\beta}_\ell + \sqrt{\tau_{\mathsf{c}}^t}\boldsymbol{Z}_\ell$, where $\boldsymbol{Z}_\ell \in \mathbb{R}^M$ is a standard normal random vector independent of $\boldsymbol{\beta}_\ell$, and the effective noise variance τ_{c}^t is given by (2.10). Under the above

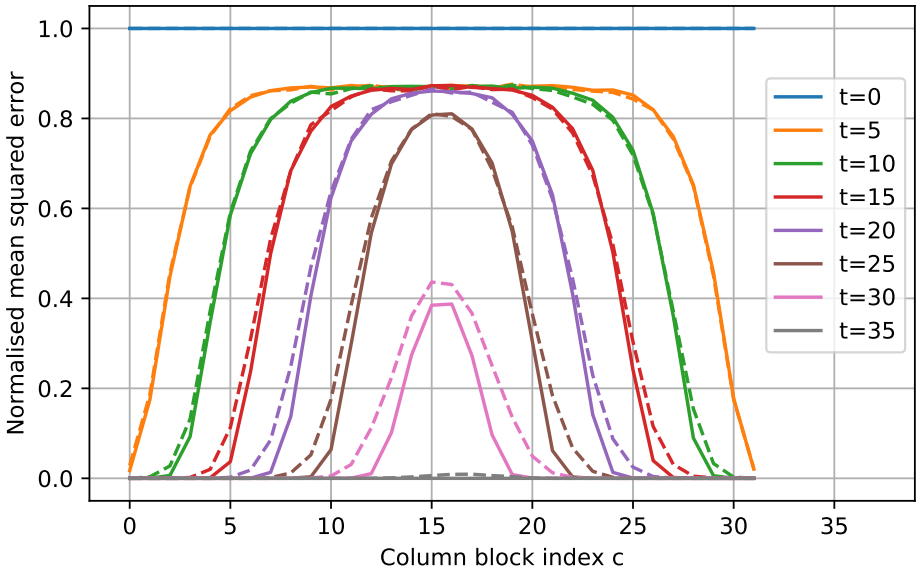


Figure 2.2: Normalised mean squared error $\frac{\|\boldsymbol{\beta}_{\mathsf{c}}^t - \boldsymbol{\beta}_{\mathsf{c}}\|_2^2}{L/\mathsf{C}}$ versus column block index $\mathsf{c} \in [\mathsf{C}]$ for several AMP decoding iterations of an SC-SPARC. The SC-SPARC is constructed with an $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix and has the following parameters: $R = 1.5$ bits, $M = 512$, $L = 2048$ and $n = 12284$. The channel capacity is $C = 2$ bits. The solid lines are the state evolution predictions from (2.10), and the dotted lines are the average NMSE over 100 instances of AMP decoding.

distributional assumption, the denoising function η_j^t in (2.14) is the minimum mean squared error (MMSE) estimator for β_j, i.e.,

$$\eta_j^t(\boldsymbol{s}^t) = \mathbb{E}\Big[\beta_j \mid \boldsymbol{\beta}_\ell + \sqrt{\tau_{\mathsf{c}}^t}\, \boldsymbol{Z}_\ell = \boldsymbol{s}_\ell^t\Big], \qquad \text{for } j \in [LM], \tag{2.15}$$

where the expectation is calculated over and $\boldsymbol{Z}_\ell \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_M)$ and $\boldsymbol{\beta}_\ell$, which is uniformly distributed over the M vectors with a single non-zero entry equal to 1. Moreover, ψ_{c}^{t+1} in (2.10) of the state evolution recursion corresponds to the MMSE value, i.e.,

$$\psi_{\mathsf{c}}^{t+1} = \mathbb{E}\Big\{\Big\|\boldsymbol{\beta}_\ell - \eta^t\Big(\boldsymbol{\beta}_\ell + \sqrt{\tau_{\mathsf{c}}^t}\, \boldsymbol{Z}_\ell\Big)\Big\|^2\Big\} = \mathbb{E}\Big\{\Big\|\boldsymbol{\beta}_\ell - \mathbb{E}\Big[\boldsymbol{\beta}_\ell \mid \boldsymbol{\beta}_\ell + \sqrt{\tau_{\mathsf{c}}^t}\, \boldsymbol{Z}_\ell\Big]\Big\|^2\Big\}. \tag{2.16}$$

Due to the above distributional property of the input vector to $\eta^t(\cdot)$, the AMP estimate of the message vector $\boldsymbol{\beta}^{t+1}$, which is the output vector of $\eta^t(\cdot)$, achieves a (block-wise) normalised mean squared error (NMSE) that can be predicted by the ψ_{c}^{t+1}'s of state evolution, i.e., $\frac{\|\beta_{\mathsf{c}}^{t+1} - \beta_{\mathsf{c}}\|_2^2}{L/\mathsf{C}} \approx \psi_{\mathsf{c}}^{t+1}$ for $\mathsf{c} \in [\mathsf{C}]$. This is illustrated in Fig. 2.2. In Fig. 2.2 we also observe that as the AMP iterates, the reduction in NMSE propagates from the blocks at the two ends towards the blocks in the middle. This phenomenon can be explained using the state evolution analysis in Section 2.3.

The vector $\boldsymbol{z}^t$ in (2.7) is a modified residual vector, consisting of the residual $\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}^t$ and an "Onsager" correction term $\overline{\boldsymbol{v}}^t \odot \boldsymbol{z}^{t-1}$. The entries of the modified residual $\boldsymbol{z}^t$ in (2.7) are approximately Gaussian and independent, with the variance determined by the block index.

For $\mathsf{r} \in [\mathsf{R}]$, the SE parameter ϕ_{r}^t approximates the variance of $\boldsymbol{z}_{\mathsf{r}}^t$, the r-th block of the residual. The Onsager term arises naturally in the derivation of the AMP algorithm, and is crucial for the Gaussian distributional properties mentioned above. For intuition about the role of the Onsager term, see [76, Sec. I-C] and [139, Sec. VI].

The key difference between the AMP decoder and state evolution for SC-SPARCs and that for uncoupled SPARCs [22] is that for SC-SPARCs, the variances of the effective observation and the residual depend on the column- and row-block indices, respectively. These variances are captured by $\{\tau_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$ and $\{\phi_{\mathsf{r}}^t\}_{\mathsf{r}\in[\mathsf{R}]}$.

2.2.1 The error performance of the AMP decoder

The performance of a SPARC decoder is measured by the *section error rate* (SER) — the fraction of sections decoded in error. It is defined as

$$\text{SER} := \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\left\{\hat{\boldsymbol{\beta}}_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)}\right\}, \tag{2.17}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, $\boldsymbol{\beta}_{\text{sec}(\ell)} \in \mathbb{R}^M$ is the ℓ-th section of the message vector, and $\hat{\boldsymbol{\beta}}_{\text{sec}(\ell)}$ is the AMP decoder's estimate of that section (after the final hard-decision step).

If the AMP decoder is run for T iterations, the section error rate can be bounded in terms of the squared error $\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2$ as follows, where $\boldsymbol{\beta}^T$ is the AMP decoder's estimate after T iterations (without the hard-decision step). Since the unique non-zero entry in any section $\ell \in [L]$ of $\boldsymbol{\beta}$ equals 1 and $\hat{\boldsymbol{\beta}}_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)}$ implies that the corresponding element of $\boldsymbol{\beta}^T_{\text{sec}(\ell)}$ is less than or equal to $1/2$,

$$\hat{\boldsymbol{\beta}}_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)} \quad \Rightarrow \quad \|\boldsymbol{\beta}^T_{\text{sec}(\ell)} - \boldsymbol{\beta}_{\text{sec}(\ell)}\|_2^2 \geq \frac{1}{4}. \tag{2.18}$$

We recall that $\boldsymbol{\beta}_{\mathsf{c}} \in \mathbb{R}^{LM/\mathsf{C}}$ corresponds to the c-th column block of the message vector. There are $\frac{L}{\mathsf{C}}$ sections in $\boldsymbol{\beta}_{\mathsf{c}}$, with the non-zero entry in each section being equal to 1; we denote by $\boldsymbol{\beta}_{\mathsf{c}_\ell}$ the ℓ-th of these sections, for $\ell \in [L/\mathsf{C}]$. Then, (2.18) implies

$$\begin{aligned}
\text{SER} &= \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\{\widehat{\boldsymbol{\beta}}_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)}\} = \frac{1}{L}\sum_{\mathsf{c}=1}^{\mathsf{C}}\sum_{\ell=1}^{L/\mathsf{C}} \mathbb{1}\left\{\widehat{\boldsymbol{\beta}}_{\mathsf{c}_\ell} \neq \boldsymbol{\beta}_{\mathsf{c}_\ell}\right\} \\
&\leq \frac{4}{L}\sum_{\mathsf{c}=1}^{\mathsf{C}}\sum_{\ell=1}^{L/\mathsf{C}} \left\|\boldsymbol{\beta}^T_{\mathsf{c}_\ell} - \boldsymbol{\beta}_{\mathsf{c}_\ell}\right\|^2 = 4\left[\frac{1}{\mathsf{C}}\sum_{\mathsf{c}=1}^{\mathsf{C}} \frac{\|\boldsymbol{\beta}^T_{\mathsf{c}} - \boldsymbol{\beta}_{\mathsf{c}}\|_2^2}{L/\mathsf{C}}\right] = 4\left[\frac{1}{L}\left\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\right\|^2\right].
\end{aligned} \tag{2.19}$$

Due to (2.19), an upper bound on the normalised mean square error (NMSE) $\left\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\right\|^2/L$ will provide an upper bound on the section error rate. Furthermore, [1, Thm. 2] proved that the NMSE concentrates on $\sum_{\mathsf{c}=1}^{\mathsf{C}} \psi_{\mathsf{c}}^T/\mathsf{C}$ for large (n, L), where $\{\psi_{\mathsf{c}}^T\}_{\mathsf{c}\in[\mathsf{C}]}$ are the state evolution parameters in (2.10). In the next section, we obtain bounds on $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$ and understand the

decoding progression of the AMP decoder by analysing the state evolution recursion.

2.3 Decoding progression according to state evolution

In this section we analyse the NMSE achieved by SC-SPARCs with AMP decoding via the state evolution recursion. We derive bounds for the state evolution parameters $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$ which lead to a succinct asymptotic characterisation of state evolution (as section size $M \to \infty$). In Section 2.3.1, we consider the asymptotic state evolution for $(\omega, \Lambda, \rho = 0)$ base matrices. Our asymptotic state evolution analysis helps to explain the decoding progression illustrated in Fig. 2.2 and why SC-SPARCs with AMP decoding are asymptotically capacity achieving. Then in Section 2.3.2, we refine our results for large but finite values of M and small but positive values of ρ.

The following lemma provides lower and upper bounds the state evolution parameters $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$ which predict the block-wise NMSE of the AMP decoder.

Lemma 2.3.1 ([1, Lem. 4.1])**.** *Let $\boldsymbol{W} \in \mathbb{R}^{R\times C}$ be a base matrix having row and column averages that are bounded above and below by strictly positive constants. That is, there exist constants $\kappa_{\mathrm{L}}, \kappa_{\mathrm{U}} > 0$ such that*

$$\kappa_{\mathrm{L}} \leq \ \frac{1}{C}\sum_{c'} W_{rc'},\ \frac{1}{R}\sum_{r'} W_{r'c} \leq \kappa_{\mathrm{U}}, \qquad r \in [R],\ c \in [C].$$

Let

$$\nu_c^t \coloneqq \frac{R}{\tau_c^t \ln M} = \frac{1}{R}\sum_{r=1}^{R} \frac{W_{rc}}{\phi_r^t}. \tag{2.20}$$

For sufficiently large M and any $\delta \in (0, \frac{1}{2})$, $\tilde{\delta} \in (0,1)$,

$$\left(1 - M^{-k_1\tilde{\delta}^2}\right) \mathbb{1}\{\nu_c^t < (2-\tilde{\delta})R\} \ \leq\ \psi_c^{t+1} \leq 1 - \left(1 - \frac{M^{-k\delta^2}}{\delta\sqrt{\ln M}}\right) \mathbb{1}\{\nu_c^t > (2+\delta)R\}, \qquad c \in [C], \tag{2.21}$$

where k, k_1 are positive constants depending only on κ_{L} and κ_{U}.

Asymptotic state evolution Noting that $\psi_{\mathsf{c}}^{t+1} \in [0,1]$, Lemma 2.3.1 implies the following asymptotic state evolution recursion as $M \to \infty$. Initialise $\bar{\psi}_{\mathsf{c}}^0 = 1$, for $\mathsf{c} \in [\mathsf{C}]$, and for $t = 0, 1, 2, \ldots$,

$$\bar{\phi}_{\mathsf{r}}^t = \sigma^2 + \frac{1}{\mathsf{C}}\sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}}\bar{\psi}_{\mathsf{c}}^t, \qquad \mathsf{r} \in [\mathsf{R}], \tag{2.22}$$

$$\bar{\psi}_{\mathsf{c}}^{t+1} = \mathbb{1}\left\{\frac{1}{\mathsf{R}}\sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\bar{\phi}_{\mathsf{r}}^t} \leq 2R\right\}, \qquad \mathsf{c} \in [\mathsf{C}], \tag{2.23}$$

where $\bar{\phi}, \bar{\psi}$ indicate asymptotic values as $M \to \infty$.

The term $\bar{\nu}_{\mathsf{c}}^t \coloneqq \frac{1}{\mathsf{R}} \sum_{\mathsf{r}} (W_{\mathsf{rc}}/\bar{\phi}_{\mathsf{r}}^t)$ in (2.23) represents the average signal to effective noise ratio for the column block index c after iteration t. If this quantity exceeds the threshold of $2R$, then the c-th column block of the message vector, $\boldsymbol{\beta}_{\mathsf{c}}$, will be decoded in the next iteration in the large system limit, i.e., $\bar{\psi}_{\mathsf{c}}^{t+1} = 0$. If we terminate the AMP decoder at iteration T, we want $\psi_{\mathsf{c}}^T = 0$, for all $\mathsf{c} \in [\mathsf{C}]$, so that the entire message vector is decoded correctly.

Remark 2.3.1 (BP threshold)**.** For rates R smaller than the belief propagation (BP) threshold

$$R_{\text{BP}} \coloneqq \frac{\mathsf{snr}}{2(1+\mathsf{snr})}, \tag{2.24}$$

where $\mathsf{snr} = P/\sigma^2$, one does not require power allocation or spatial coupling for reliable SPARC decoding. Indeed, consider a standard uncoupled SPARC where the 1-by-1 base matrix is a single entry equal to P. Using the asymptotic state evolution recursion (2.22)–(2.23), we see that if $R < R_{\text{BP}}$, then the whole message vector decodes in one iteration, i.e., $\bar{\psi}^1 = 0$.

2.3.1 Asymptotic analysis

The asymptotic SE recursion (2.22)–(2.23) is given for a general base matrix $\boldsymbol{W}$. To get some insight into the decoding progression, we specialise the result to the $(\omega, \Lambda, \rho = 0)$ base matrix introduced in Definition 2.1.1. Recall that an $(\omega, \Lambda, \rho = 0)$ base matrix has $\mathsf{R} = \Lambda + \omega - 1$ rows and $\mathsf{C} = \Lambda$ columns, with each column having ω non-zero entries each equal to $P \cdot \frac{\Lambda+\omega-1}{\omega}$.

Corollary 2.3.1. *The asymptotic state evolution recursion* (2.22)–(2.23) *for an* $(\omega, \Lambda, \rho = 0)$ *base matrix is as follows. Initialise* $\bar{\psi}_{\mathsf{c}}^0 = 1$*, for* $\mathsf{c} \in [\Lambda]$*, and for* $t = 0, 1, 2, \ldots,$

$$\bar{\phi}_{\mathsf{r}}^t = \sigma^2 + \frac{\vartheta\, P}{\omega} \sum_{\mathsf{c}=\underline{\mathsf{c}}_{\mathsf{r}}}^{\bar{\mathsf{c}}_{\mathsf{r}}} \bar{\psi}_{\mathsf{c}}^t, \qquad \mathsf{r} \in [\Lambda + \omega - 1], \tag{2.25}$$

$$\bar{\nu}_{\mathsf{c}}^t = \frac{P}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{\mathsf{c}+\omega-1} \frac{1}{\bar{\phi}_{\mathsf{r}}^t}, \qquad \bar{\psi}_{\mathsf{c}}^{t+1} = \mathbb{1}\{\bar{\nu}_{\mathsf{c}}^t \le 2R\}, \qquad \mathsf{c} \in [\Lambda], \tag{2.26}$$

where $\vartheta = \frac{\Lambda+\omega-1}{\Lambda}$*, and*

$$(\underline{\mathsf{c}}_{\mathsf{r}},\, \bar{\mathsf{c}}_{\mathsf{r}}) = \begin{cases} (1,\, \mathsf{r}) & \text{if } \ 1 \le \mathsf{r} \le \omega \\ (\mathsf{r}-\omega+1,\, \mathsf{r}) & \text{if } \ \omega \le \mathsf{r} \le \Lambda \\ (\mathsf{r}-\omega+1,\, \Lambda) & \text{if } \ \Lambda \le \mathsf{r} \le \Lambda+\omega-1. \end{cases} \tag{2.27}$$

Proof. Substitute the value of W_{rc} from (2.4), with $\rho = 0$ and $\mathsf{C} = \Lambda$, $\mathsf{R} = \Lambda + \omega - 1$ in (2.22)–(2.23). □

Observe that the $\bar{\phi}_{\mathsf{r}}^t$'s, $\bar{\nu}_{\mathsf{c}}^t$'s, and $\bar{\psi}_{\mathsf{c}}^t$'s are symmetric about the middle indices, i.e., $\bar{\phi}_{\mathsf{r}}^t = \bar{\phi}_{\mathsf{R}-\mathsf{r}+1}^t$

for $\mathsf{r} \leq \lfloor \frac{\mathsf{R}}{2} \rfloor$ and $\bar{\nu}_{\mathsf{c}}^{t} = \bar{\nu}_{\mathsf{C}-\mathsf{c}+1}^{t}$, $\bar{\psi}_{\mathsf{c}}^{t} = \bar{\psi}_{\mathsf{C}-\mathsf{c}+1}^{t}$ for $\mathsf{c} \leq \lfloor \frac{\mathsf{C}}{2} \rfloor$.

Decoding initialisation

Consider the initial step ($t = 0$): from (2.25) the value of $\bar{\phi}_{\mathsf{r}}^{0}$ depends on the number of non-zero entries in row r of $\boldsymbol{W}$, which is equal to $\overline{\mathsf{c}}_{\mathsf{r}} - \underline{\mathsf{c}}_{\mathsf{r}} + 1$, with $\overline{\mathsf{c}}_{\mathsf{r}}, \underline{\mathsf{c}}_{\mathsf{r}}$ given by (2.27). Therefore, $\bar{\phi}_{\mathsf{r}}^{0}$ increases from $\mathsf{r} = 1$ until $\mathsf{r} = \omega$, is constant for $\omega \leq \mathsf{r} \leq \Lambda$, and then starts decreasing again for $\Lambda < \mathsf{r} \leq \Lambda + \omega - 1$. As a result, $\bar{\psi}_{\mathsf{c}}^{1}$ is smallest for c at either end of the base matrix ($\mathsf{c} \in \{1, \Lambda\}$) and increases as c moves towards the middle, since the $\bar{\nu}_{\mathsf{c}}^{0} = \sum_{\mathsf{r}=\mathsf{c}}^{\mathsf{c}+\omega-1} (\bar{\phi}_{\mathsf{r}}^{0})^{-1}$ term in (2.26) is largest for $\mathsf{c} \in \{1, \Lambda\}$, followed by $\mathsf{c} \in \{2, \Lambda - 1\}$, and so on. Therefore, we expect the blocks of the message vector corresponding to column block index $\mathsf{c} \in \{1, \Lambda\}$ to be decoded most easily, followed by $\mathsf{c} \in \{2, \Lambda - 1\}$, and so on. Fig. 2.2 shows that this is indeed the case as the blocks of the message vector at the two ends decode first.

Rates below the BP threshold Since the noise variance $\bar{\phi}_{\mathsf{r}}^{0}$ is highest (and constant) for $\omega \leq \mathsf{r} \leq \Lambda$, the average signal to noise ratio $\bar{\nu}_{\mathsf{c}}^{0}$ is lowest (and constant) for column block indices $\omega \leq \mathsf{c} \leq \Lambda - \omega + 1$. Therefore, all column blocks of the message vector to decode in the first iteration ($\bar{\psi}_{\mathsf{c}}^{1} = 0$ for $\mathsf{c} \in [\Lambda]$) if $\bar{\nu}_{\omega}^{0} > 2R$. We now obtain a lower bound on $\bar{\nu}_{\omega}^{0}$ to get a sufficient condition for all column blocks to decode in the first iteration.

$$\bar{\nu}_{\omega}^{0} = \frac{\mathsf{snr}}{\omega} \sum_{\mathsf{r}=\omega}^{2\omega-1} \frac{1}{1 + \frac{\vartheta\,\mathsf{snr}}{\omega}(\overline{\mathsf{c}}_{\mathsf{r}} - \underline{\mathsf{c}}_{\mathsf{r}} + 1)} \geq \frac{\mathsf{snr}}{1 + \vartheta\,\mathsf{snr}}, \tag{2.28}$$

where $\mathsf{snr} = P/\sigma^2$. The inequality holds because $(\overline{\mathsf{c}}_{\mathsf{r}} - \underline{\mathsf{c}}_{\mathsf{r}} + 1) \leq \omega$ from (2.27), and becomes an equality if $\Lambda \geq 3\omega - 2$. Therefore, if the rate satisfies

$$R < \frac{\mathsf{snr}}{2(1 + \vartheta\,\mathsf{snr})}, \tag{2.29}$$

then all column blocks of the message vector to decode in the first iteration. Note that $\frac{\mathsf{snr}}{2(1+\vartheta\,\mathsf{snr})} \leq R_{\mathrm{BP}}$, and recall that for rates less than R_{BP}, there is no need for spatial coupling (see Remark 2.3.1).

Rates above the BP threshold For rates greater than R_{BP} and $\Lambda \geq 3\omega - 2$, we know that it is impossible for all column blocks of the message decoder to decode in the first iteration by using the same arguments as above. In order for *any* column block to decode, we require $\bar{\nu}_{1}^{0} > 2R$ since $\bar{\nu}_{\mathsf{c}}^{0}$ is largest for $\mathsf{c} \in \{1, \Lambda\}$ and $\bar{\nu}_{1}^{0} = \bar{\nu}_{\Lambda}^{0}$ due to symmetry. We now obtain a lower bound on $\bar{\nu}_{1}^{0}$ to get a sufficient condition for decoding to start. For column block index $\mathsf{c} < \omega$, we have

$$\bar{\nu}_{\mathsf{c}}^{0} = \frac{\mathsf{snr}}{\omega}\left(\sum_{\mathsf{r}=\mathsf{c}}^{\omega-1} \frac{1}{1 + \frac{\vartheta\,\mathsf{snr}}{\omega}\,\mathsf{r}} + \frac{\mathsf{c}}{1 + \vartheta\,\mathsf{snr}} \right) \overset{\text{(i)}}{>} \frac{1}{\vartheta} \int_{\frac{\vartheta\,\mathsf{snr}}{\omega}\mathsf{c}}^{\frac{\vartheta\,\mathsf{snr}}{\omega}\omega} \frac{1}{1+x}\,dx + \frac{\mathsf{c}}{\omega}\frac{\mathsf{snr}}{(1 + \vartheta\,\mathsf{snr})}$$

$$= \frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \frac{1}{\vartheta}\ln\left(1+\vartheta\,\mathsf{snr}\,\frac{\mathsf{c}}{\omega}\right) + \frac{\mathsf{c}}{\omega}\frac{\mathsf{snr}}{(1+\vartheta\,\mathsf{snr})}$$
$$\overset{\text{(ii)}}{>} \frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \frac{\mathsf{c}}{\omega}\,\mathsf{snr} + \frac{\mathsf{c}}{\omega}\frac{\mathsf{snr}}{(1+\vartheta\,\mathsf{snr})} = \frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \frac{\mathsf{c}}{\omega}\frac{\vartheta\,\mathsf{snr}^2}{(1+\vartheta\,\mathsf{snr})}, \tag{2.30}$$

where (i) is obtained by using a definite integral to lower bound the left Riemann sum of the positive decreasing function $\frac{1}{1+x}$, and (ii) from $\ln x \leq x-1$. Therefore, a sufficient condition for $\bar{\nu}_1^0 > 2R$ is

$$R < \frac{1}{2\vartheta}\ln(1+\vartheta\,\mathsf{snr}), \tag{2.31}$$
$$\omega > \frac{\vartheta\,\mathsf{snr}^2}{2(1+\vartheta\,\mathsf{snr})}\left(\frac{1}{2\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - R\right)^{-1}. \tag{2.32}$$

If the above conditions are satisfied, then (at least) the first and last column block of the message vector decodes, i.e., $\bar{\psi}_1^1 = \bar{\psi}_\Lambda^1 = 0$. Note that the right-hand-side (RHS) of (2.31) can be made arbitrarily close to the channel capacity C by making $\frac{\omega-1}{\Lambda}$ small enough (recall that $\vartheta = 1 + \frac{\omega-1}{\Lambda}$). Indeed, since the RHS of (2.31) is decreasing in ϑ for $\vartheta > 1$, we have

$$C > \frac{1}{2\vartheta}\ln(1+\vartheta\,\mathsf{snr}) > \frac{C}{\vartheta}. \tag{2.33}$$

Therefore, the term in the brackets in (2.32) can be seen as a rate gap to capacity term, and the requirement on the coupling width ω is inversely proportional to this gap.

Remark 2.3.2 (Choice of base matrix parameters)**.** For any fixed rate $R < C = \frac{1}{2}\ln(1+\mathsf{snr})$, the base matrix parameters (ω, Λ) can be chosen such that the conditions in Eqs. (2.31)–(2.32) are satisfied. Indeed, consider a rate $R = C/\vartheta_0$, for any constant $\vartheta_0 > 1$. Then choose ω to satisfy (2.32) (with ϑ replaced by ϑ_0), and Λ large enough that $\vartheta = \frac{\Lambda+\omega-1}{\Lambda} \leq \vartheta_0$. Therefore, for any fixed rate $R < C$ and suitably chosen base matrix parameters (ω, Λ), the AMP decoder can start successfully decoding the message vector in the limit of large (n, L, M) (whilst satisfying $nR = L\ln M$).

Decoding progression

From the "Decoding initialisation" section above, we know that if $R < R_{\text{BP}}$, then all column blocks of the message vector decode in one iteration. In this section we explain the decoding propagation phenomenon seen in Fig. 2.2 for rates between the BP threshold and the channel capacity C, assuming that $\Lambda \geq 3\omega - 2$.

From the section above, we know that if $R > R_{\text{BP}}$, then fewer than ω column blocks (from each end) of the message vector decode in the first iteration. Furthermore, if $R < C$ and the base matrix parameters (ω, Λ) are appropriately chosen (Remark 2.3.2), then at least one column block (from each end) of the message vector decodes in the first iteration. We now find

all the column block indices $\mathsf{c} < \omega$ that decode in the first iteration for these choices of R and (ω, Λ), i.e., the indices $\mathsf{c} < \omega$ for which $\bar{\psi}_{\mathsf{c}}^1 = 0$, or equivalently $\bar{\nu}_{\mathsf{c}}^0 > 2R$. Using the lower bound on $\bar{\nu}_{\mathsf{c}}^0$ for $\mathsf{c} < \omega$ given in (2.30), a sufficient condition for $\bar{\nu}_{\mathsf{c}}^0 > 2R$ is

$$\mathsf{c} < \bar{g} \coloneqq \omega \cdot \frac{2(1+\vartheta\,\mathsf{snr})}{\vartheta\,\mathsf{snr}^2}\bigg(\frac{1}{2\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - R\bigg). \tag{2.34}$$

Therefore, all column blocks whose index satisfies $\mathsf{c} < \bar{g}$ (and $\mathsf{c} > \Lambda - \bar{g} + 1$) will decode in the first iteration, i.e., $\bar{\psi}_{\mathsf{c}}^1 = 0$. Note that $1 < \bar{g} \leq \omega$ due to the condition on ω in (2.32) and because $R > R_{\mathrm{BP}} > \frac{\mathsf{snr}}{2(1+\vartheta\,\mathsf{snr})}$.

We next consider subsequent iterations $t > 1$. Assume towards induction that

$$\bar{\psi}_{\mathsf{c}}^t = \bar{\psi}_{\Lambda-\mathsf{c}+1}^t = 0 \quad \text{for } \mathsf{c} < t\bar{g}, \tag{2.35}$$

We will prove that (2.35) implies $\bar{\psi}_{\mathsf{c}}^{t+1} = \bar{\psi}_{\Lambda-\mathsf{c}+1}^{t+1} = 0$ for $\mathsf{c} < (t+1)\bar{g}$, i.e., in each iteration at least $\lfloor \bar{g} \rfloor$ additional column blocks from each end of the message vector decode. We find the column block indices $\mathsf{c} \in (t\bar{g}, t\bar{g} + \omega)$ for which $\bar{\psi}_{\mathsf{c}}^{t+1} = 0$, or equivalently $\bar{\nu}_{\mathsf{c}}^t > 2R$. The same analysis applies to the indices at the other end due to symmetry. Using the induction assumption (2.35) in the asymptotic state evolution (2.25)–(2.27), we have

$$\frac{\bar{\phi}_r^t}{\sigma^2} \leq \begin{cases} 1, & \mathsf{r} \leq t\bar{g}, \\ 1 + \frac{\vartheta\,\mathsf{snr}}{\omega}(\mathsf{r} - t\bar{g}), & t\bar{g} < \mathsf{r} < t\bar{g} + \omega, \\ 1 + \vartheta\,\mathsf{snr}, & \mathsf{r} \geq t\bar{g} + \omega. \end{cases} \tag{2.36}$$

We now obtain a lower bound on $\bar{\nu}_{\mathsf{c}}^t$ for $\mathsf{c} \in (t\bar{g}, t\bar{g} + \omega)$ using the same steps as (2.30).

$$\begin{aligned} \bar{\nu}_{\mathsf{c}}^t &\geq \frac{\mathsf{snr}}{\omega}\bigg(\sum_{\mathsf{r}=\mathsf{c}}^{t\bar{g}+\omega-1} \frac{1}{1 + \frac{\vartheta\,\mathsf{snr}}{\omega}(\mathsf{r} - t\bar{g})} + \frac{\mathsf{c} - t\bar{g}}{1+\vartheta\,\mathsf{snr}}\bigg) = \frac{\mathsf{snr}}{\omega}\bigg(\sum_{\mathsf{r}=\mathsf{c}-t\bar{g}}^{\omega-1} \frac{1}{1 + \frac{\vartheta\,\mathsf{snr}}{\omega}\mathsf{r}} + \frac{\mathsf{c} - t\bar{g}}{1+\vartheta\,\mathsf{snr}}\bigg) \\ &> \frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \frac{\mathsf{c} - t\bar{g}}{\omega}\frac{\vartheta\,\mathsf{snr}^2}{(1+\vartheta\,\mathsf{snr})}. \end{aligned} \tag{2.37}$$

Therefore, for column block indices $\mathsf{c} \in (t\bar{g}, t\bar{g} + \omega)$, a sufficient condition for $\bar{\psi}_{\mathsf{c}}^{t+1} = 0$, or equivalently $\bar{\nu}_{\mathsf{c}}^t > 2R$, is $\mathsf{c} < (t+1)\bar{g}$. We thus conclude the proof by induction.

The above analysis implies that for rates satisfying $R_{\mathrm{BP}} < R < C$ and appropriately chosen base matrix parameters, all column blocks of the message vector decode in at most $\lceil \Lambda/(2\bar{g}) \rceil$ iterations. Since $\bar{g}$ is proportional to the 'rate gap to capacity' term $\frac{1}{2\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - R$, this upper bound on the number decoding iterations increases as the rate approaches the channel capacity.

Remark 2.3.3 (Power allocated SPARCs)**.** One can analyse the decoding progression of AMP decoded power allocated SPARCs in a similar way by substituting the corresponding base

matrix of the power allocation scheme into the asymptotic state evolution (2.22)–(2.23). See [22, Lemma 2] for the asymptotic state evolution analysis of the decoding progression of a SPARC with exponentially decaying power allocation (2.3).

Decoding velocity The wave-like decoding propagation in Fig. 2.2 has a roughly constant decoding velocity, i.e., roughly the same number of column blocks decode every 5 iterations until the two decoding waves "merge" and "collapse" near the middle. In our analysis above, we showed that at least $\lfloor \bar{g} \rfloor$ additional column blocks from each end of the message vector decode in each iteration. In order to prove the decoding velocity is roughly constant, we require an upper bound that is of the same order as $\bar{g}$. We expect this to be the case, but have not yet proved it.

This wave-like decoding propagation also occurs in spatially coupled LDPC codes decoded with belief propagation. The propagation of the LDPC decoding wave (in the large system limit) was studied in [153].

2.3.2 Non-asymptotic analysis

In Section 2.3.1, we analysed the decoding progression of the AMP decoder for SC-SPARCs based on $(\omega, \Lambda, \rho = 0)$ base matrices in limit of $M \to \infty$. In this section, we consider the decoding progression for (ω, Λ, ρ) base matrices where ρ can be a small positive value, and refine the previous analysis to account for large but finite values of M using Lemma 2.3.1. This non-asymptotic result (Proposition 2.3.1) will be used to establish the result that SC-SPARCs with AMP decoding are asymptotically capacity achieving in the next section.

We need to analyse the $\rho > 0$ case because the analysis that shows the NMSE of the AMP decoder concentrates around the state evolution prediction only holds for base matrices with entries that are lower bounded by a strictly positive constant [1, Thm. 2].

Proposition 2.3.1. *Consider a rate R SC-SPARC with an $n \times LM$ design matrix constructed using an (ω, Λ, ρ) base matrix and consider a constant $\delta \in (0, \min\{\frac{\Delta}{2R}, \frac{1}{2}\})$, where $0 \leq \rho \leq \min\{\frac{\Delta}{3\mathsf{snr}}, \frac{1}{2}\}$, and*

$$\Delta := \frac{1}{2\vartheta} \ln(1 + \vartheta \mathsf{snr}) - R. \tag{2.38}$$

If the rate satisfies $R < \frac{(1-\rho)\mathsf{snr}}{(2+\delta)(1+\vartheta\mathsf{snr})}$, then all the column blocks of the message vector simultaneously decode in one iteration, i.e., for all $\mathsf{c} \in [\Lambda]$,

$$\psi_{\mathsf{c}}^{1} \leq f_{M,\delta} := \frac{M^{-k\delta^2}}{\delta\sqrt{\ln M}} \tag{2.39}$$

for sufficiently large M, where $k > 0$ is a universal constant.

Otherwise, if the rate satisfies $\frac{(1-\rho)\mathsf{snr}}{(2+\delta)(1+\vartheta\mathsf{snr})} \leq R < \frac{1}{2\vartheta} \ln(1 + \vartheta\, \mathsf{snr})$, and the coupling width ω

satisfies

$$\omega > \left(\frac{\vartheta\, \mathsf{snr}^2}{1+\vartheta \mathsf{snr}}\right)\frac{1}{\Delta}, \tag{2.40}$$

then, for $t \geq 1$ *and*

$$\mathsf{c} \leq \max\left\{tg,\ \left\lceil\frac{\Lambda}{2}\right\rceil\right\}, \tag{2.41}$$

where

$$g = \frac{(1+\vartheta\, \mathsf{snr})\Delta}{\vartheta\, \mathsf{snr}^2}\,\omega, \tag{2.42}$$

we have

$$\psi_{\mathsf{c}}^t \;=\; \psi_{\Lambda-\mathsf{c}+1}^t \;\leq\; f_{M,\delta} \tag{2.43}$$

for sufficiently large M*.*

The proof of this proposition is given in Section 2.7.

Decoding progression

The discussions made in Section 2.3.1 concerning decoding initialisation and decoding progression also hold for the non-asymptotic setting considered here with slight modifications.

For example, for rates larger than the BP threshold, the proposition says that if the coupling width ω is large enough (as specified by (2.40)), then in iteration t at least the first and last $\lfloor tg \rfloor$ column blocks from each end of the message vector are expected to decode, i.e., have their state evolution predicted NMSE less than or equal to $f_{M,\delta}$. Furthermore, the proof shows that if $g_t \leq \Lambda/2$ is the exact number of column blocks such that $\psi_{\mathsf{c}}^t = \psi_{\Lambda-\mathsf{c}+1}^t \leq f_{M,\delta}$ for $\mathsf{c} \leq g_t$, then $g_{t+1} \geq \lfloor g_t + g \rfloor$, i.e., in each iteration at least $\lfloor g \rfloor \geq 1$ additional column blocks of the message vector from each end are expected to decode.

This decoding progression continues until iteration T when all column blocks have been decoded, i.e., $\psi_{\mathsf{c}}^T \leq f_{M,\delta}$ for $\mathsf{c} \in [\Lambda]$. More precisely, we run the AMP decoder for T iterations where

$$T := \min\{t :\ \psi_{\mathsf{c}}^t \leq f_{M,\delta} \text{ for } \mathsf{c} \in [\Lambda]\}. \tag{2.44}$$

Proposition 2.3.1 implies that for rates larger than the BP threshold

$$T \leq \left\lceil \frac{\Lambda}{2g} \right\rceil. \tag{2.45}$$

We note that g is proportional to Δ, which represents the rate gap to capacity (see (2.33), (2.38)). Therefore, from (2.45) the number of iterations T grows as the rate approaches the channel capacity. For a fixed R the quantity $f_{M,\delta}$ tends to 0 with growing M.

2.4 Probability of excess section error rate

Proposition 2.3.1 shows that for any $R < C$, suitably chosen base matrix parameters (ω, Λ, ρ), and sufficiently large M, the state evolution predicted NMSE achieved by AMP decoding after a finite number of iterations can be upper bounded by a term $(f_{M,\delta})$ that can be made arbitrarily small by increasing M. Using Proposition 2.3.1 with the the result in [1, Thm. 2] where the NMSE of the AMP decoder is shown to concentrate on the state evolution estimates, and the fact that NMSE provides an upper bound on the section error rate (see (2.19)), we obtain a bound on the probability of the section error rate exceeding some fixed $\epsilon \in (0, 1)$.

Theorem 1. *For any rate satisfying $R_{BP} \leq R < C$, let $\boldsymbol{W} \in \mathbb{R}_+^{R \times C}$ be an (ω, Λ, ρ) base matrix with parameters chosen such that $R < \frac{1}{2\vartheta} \ln(1 + \vartheta\, \mathsf{snr})$, where $\vartheta = 1 + \frac{\omega - 1}{\Lambda}$, and ω is large enough that the condition in (2.40) is satisfied, and $\rho = \min\{\frac{\Delta}{3\mathsf{snr}}, \frac{1}{2}\}$, where Δ is defined in (2.38). Let $\mathcal{S}_n$ be an SC-SPARC of rate R defined via an $n \times LM$ design matrix constructed using the base matrix $\boldsymbol{W}$. The parameters (n, L, M) satisfy $nR = L \ln M$.*

Fix $\epsilon \in (0, 1)$, and for $f_{M,\delta}$ defined in (2.39), let M be large enough such that $f_{M,\delta} \leq \frac{\epsilon}{8}$ for $\delta = \min\{\frac{\Delta}{3R}, \frac{1}{3}\}$. Then the section error rate of the AMP decoder after T iterations, with T defined in (2.44), satisfies

$$\mathbb{P}\left(\mathrm{SER}(\mathcal{S}_n) > \epsilon\right) \leq K_{T-1}(R\mathsf{C})^T \exp\left(\frac{-\kappa_{T-1} n \epsilon^2}{64 (\ln M)^{2T} (R/\omega)^{2T-1}}\right). \tag{2.46}$$

For $t \geq 0$, the constants κ_t and K_t are given by $\kappa_t = [\xi^{2t}(t!)^{24}]^{-1}$ and $K_t = \Xi^{2t}(t!)^{14}$ where $\xi, \Xi > 0$ are universal constants (not depending on the AMP parameters (L, M, n, R, C) or ϵ), but not explicitly specified.

Proof. Without loss of generality, we can assume that rate gap Δ (defined in (2.38)) satisfies $\Delta < 2R$. Otherwise the arguments below hold with (Δ/R) replaced by 1.

For the choice of base matrix parameters (ω, Λ, ρ) stated in Theorem 1, we have that $\frac{1}{\mathsf{C}} \sum_{\mathsf{c}} \psi_{\mathsf{c}}^T \leq f_{M,\Delta/(3R)}$ (from Proposition 2.3.1). The AMP concentration result in [1, Thm. 2] implies that for any $\tilde{\epsilon} > 0$,

$$\mathbb{P}\left(\frac{\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2}{L} \geq \tilde{\epsilon} + f_{M,\frac{\Delta}{3R}}\right) \leq K_{T-1}(\mathsf{RC})^T \exp\left(\frac{-\kappa_{T-1} n \tilde{\epsilon}^2}{(\ln M)^{2T} (\mathsf{R}/\omega)^{2T-1}}\right). \tag{2.47}$$

Furthermore, from (2.19) we have

$$\mathbb{P}(\mathrm{SER}(\mathcal{S}_n) > \epsilon) \leq \mathbb{P}\left(\frac{\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2}{L} \geq \frac{\epsilon}{4}\right). \tag{2.48}$$

Combining (2.47) and (2.48), and taking $\tilde{\epsilon} = \frac{\epsilon}{8}$ and M large enough so that $f_{M,\Delta/(3R)} \leq \frac{\epsilon}{8}$ (see (2.39)) yields the theorem.[2] □

[2] I would like to emphasize that my contribution to the proof of Theorem 1 is the state evolution analysis in

Remark 2.4.1. Theorem 1 implies that for any fixed $R < C$ and $\epsilon \in (0, 1)$, one can construct a sequence of rate R spatially coupled SPARCs $\{\mathcal{S}_n\}$ (indexed by code length n) for which

$$\lim_{n\to\infty} \text{SER}(\mathcal{S}_n) = 0 \quad \text{almost surely.} \tag{2.49}$$

Indeed, once (Λ, ω, M) are chosen to satisfy the conditions in Theorem 1, the bound in (2.46) decreases exponentially in n. The Borel-Cantelli lemma then yields the asymptotic result in (2.49).

Remark 2.4.2 (Below the BP threshold)**.** Theorem 1 is stated for rates satisfying $R_{\text{BP}} \leq R < C$ as this is the region where spatial coupling is required. Indeed, for $R < R_{\text{BP}}$, Proposition 2.3.1 and the proof of Theorem 1 imply that the probability bound (2.46) holds with $\mathsf{R} = \mathsf{C} = \omega = 1$ and $T = 1$. This result also follows from the analysis in [57], applied with a "flat" power allocation.

2.5 Empirical performance

In this section, we investigate the finite length error performance of SC-SPARCs with AMP decoding via numerical simulations. We use the $(\omega, \Lambda, \rho = 0)$ base matrix construction in all the simulations. The implementation details will be given after the discussion of the simulation results.

SC versus non-SC SPARCs In Fig. 2.3 we compare the average section error rate (SER) of SC-SPARCs with standard (non-SC) SPARCs. The SC-SPARCs are constructed with an $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix. The standard SPARCs either have "flat" power allocation (red) or power allocation optimised using the *iterative power allocation* algorithm proposed in [47] (green). The signal-to-noise ratio $P/\sigma^2 = 15$ (capacity $C = 2$ bits), section size $M = 512$, and number of sections $L = 1024$ are fixed while the rate of the codes are varied by varying the code length n. For each rate, the code length is the same for all three codes, and AMP decoding is used for all codes.

Comparing standard SPARCs with "flat" power allocation (PA) to SC-SPARCs, we see that spatial coupling significantly improves the error performance: the rate threshold below which the SER drops steeply to a small value is a lot higher for SC-SPARCs. Comparing standard SPARCs with iterative PA to SC-SPARCs, we see that at rates close to the capacity, standard SPARCs with iterative PA have lower SER than SC-SPARCs. However, as the rate decreases, the drop in SER for standard SPARCs with iterative PA is not as steep as that for SC-SPARCs.

Section 2.3. The AMP concentration result in (2.47) was done by Ramji Venkataramanan and Cynthia Rush, and will not be discussed in this thesis.

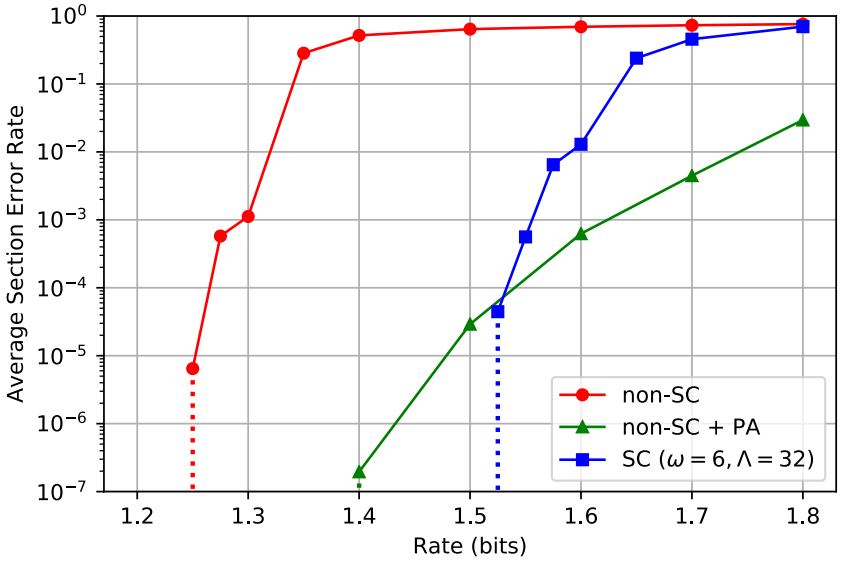


Figure 2.3: Average SER versus rate at snr $= 15$, $C = 2$ bits, $M = 512$, $L = 1024$, and $n \in [5100, 7700]$. The SERs are averaged over 10^4 trials. Plots are shown for standard (non-SC) SPARCs with "flat" (red) and iterative (green) power allocation, and SC-SPARCs constructed with an $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix (blue). For each rate, the code length is the same for all three codes, and AMP decoding is used for all codes. The dotted vertical lines indicate that no section errors were observed over 10^4 trials at smaller rates.

Effect of the coupling width ω In Fig. 2.4, we investigate the effect of the coupling width ω on the error performance. Fig. 2.4 compares the average SER of SC-SPARCs with $(\omega, \Lambda = 32, \rho = 0)$ base matrices and varying ω. For a fixed Λ, we observe from (2.5) that a larger ω results in a larger inner SPARC rate R_{inner} when the overall SC-SPARC rate R is fixed. A larger value of R_{inner} makes decoding harder; on the other hand, increasing the coupling width ω helps to jumpstart AMP decoding (Remark 2.1.1). Thus for a fixed rate, there is a trade-off as illustrated in Fig. 2.4: increasing ω improves the SER performance up to a certain point, but the performance degrades for larger ω. In general, ω should be large enough so that coupling can benefit decoding, but not so large that R_{inner} is very close to the channel capacity. For example, for $R = 1.6$ bits and $\Lambda = 32$, the inner SPARC rate $R_{\text{inner}} = 1.65, 1.75, 1.85, 1.95$ bits for $\omega = 2, 4, 6, 8$, respectively. With the capacity $C = 2$ bits, Fig. 2.4 shows that $\omega = 6$ is the best choice for $R = 1.6$ bits, with $\omega = 8$ being noticeably worse. This also indicates that smaller values ω would be favoured as the rate R gets closer to C.

SC-SPARCs versus coded modulation We now compare the performance of SC-SPARCs with that of standard coded modulation schemes such as LDPC codes with quadrature amplitude modulation (QAM). Since these standard schemes produce complex-valued symbols, we consider communication over the *complex* AWGN channel, where the noise is circularly-symmetric complex Gaussian. A *complex* SC-SPARC is defined in the same way as a real-valued SC-SPARC (Section 2.1), except the design matrix has independent circularly-symmetric com-

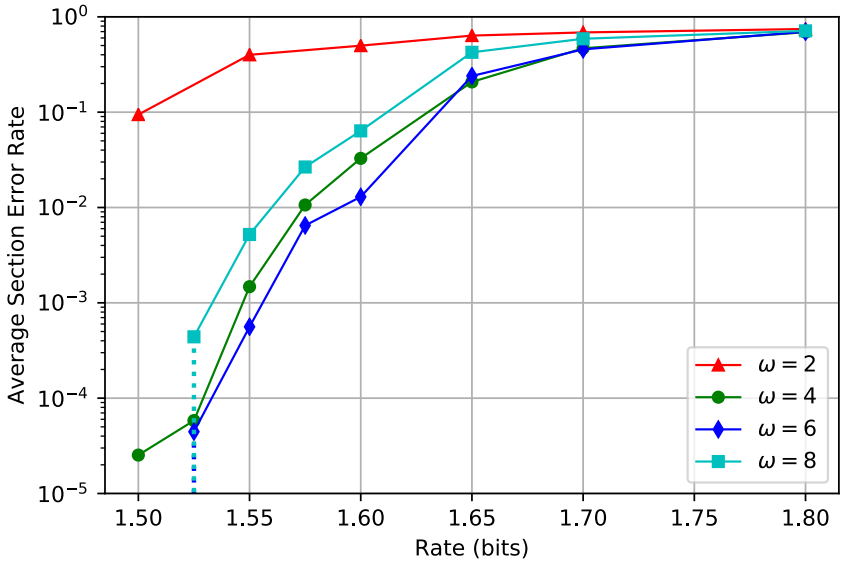


Figure 2.4: Average SER versus rate at snr $= 15$, $C = 2$ bits, $M = 512$, $L = 1024$, and $n \in [5100, 6200]$. The SERs are averaged over 10^4 trials. Plots are shown for SC-SPARCs with an $(\omega, \Lambda = 32, \rho = 0)$ base matrix and coupling width ω taking values in $\{2, 4, 6, 8\}$. For each rate, the code length is the same for different ω values. The dotted vertical line indicates that for $\omega = 6$ and 8, no section errors were observed over 10^4 trials at $R = 1.5$ bits.

plex Gaussian entries instead of real-valued Gaussian entries. The AMP decoder for complex SC-SPARCs is similar to the one in (2.7)–(2.8): we take $\boldsymbol{A}^*$ to be the conjugate transpose of $\boldsymbol{A}$, and modify the definition of η_j^t in (2.14) according to (2.15). Complex SC-SPARCs and its AMP decoder are described in detail in Chapter 3.

In Figures 2.5, 2.6, and 2.7, we provide numerical simulation results demonstrating the finite length error performance of complex SC-SPARCs with AMP decoding at different code rates and code lengths. The error performance is evaluated using both the bit error rate (BER) and the frame error rate (FER). (The FER is the message/codeword error rate.) We also simulate and plot the error performance of coded modulation schemes (LDPC + QAM) for reference using the AFF3CT toolbox [154]. The LDPC codes are chosen from the DVB-S2 standard [151] and a belief propagation (BP) decoder is used which runs for 50 iterations. For fair comparison, in each figure, the frame length of the coded modulation scheme is chosen to be close to the code length of the SC-SPARC.

Fig. 2.5 shows the performance of SC-SPARCs with rate 1.5 bits/dimension and code length $n = 10795$. The AMP decoder for the SC-SPARC is run for a maximum of 200 iterations (details in Sec. 2.5.1). The coded modulation scheme uses a rate $\frac{1}{2}$ (32400, 64800) DVB-S2 LDPC code with 64-QAM modulation, for the same overall rate of 1.5 bits/dimension and a frame length of 10800 symbols. We observe that the SC-SPARC requires a smaller E_b/N_0 to achieve BERs in the range 10^{-1} to 10^{-5}, and FERs down to 5×10^{-4} compared to the coded modulation scheme. However, for much lower FERs, we expect the coded modulation scheme to require a smaller E_b/N_0 because its frame error rate drops faster as E_b/N_0 increases.

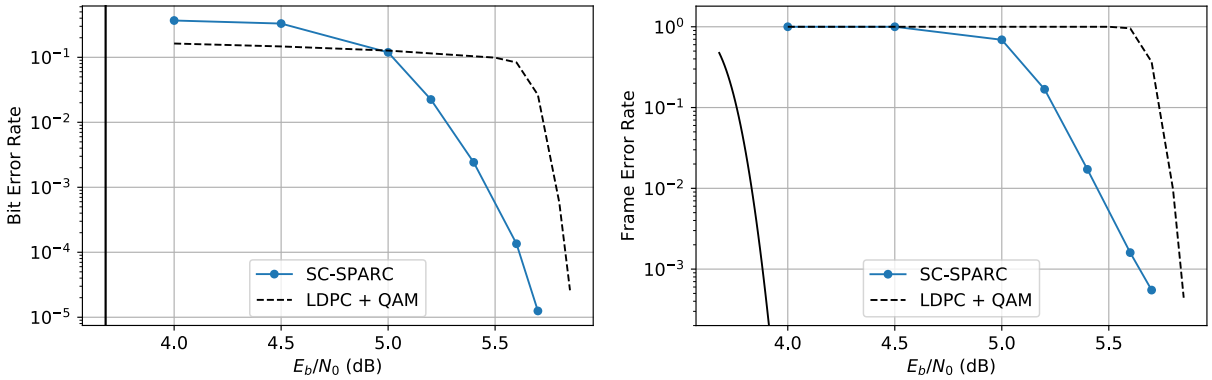


Figure 2.5: Error performance of complex SC-SPARCs defined via a $(\omega = 4, \Lambda = 32, \rho = 0)$ base matrix. Code parameters: $R = 1.5$ bits/dimension, $L = 2944$, $M = 2048$, code length $n = 10795$. The dashed lines show the performance of coded modulation: $(K = 32400, N = 64800)$ DVB-S2 LDPC + 64 QAM, frame length = 10800 symbols. The solid black line in the BER plot is the AWGN Shannon limit for $R = 1.5$ bits/dimension, and in the FER plot, it is the normal approximation to the AWGN finite length error probability bound in [34].

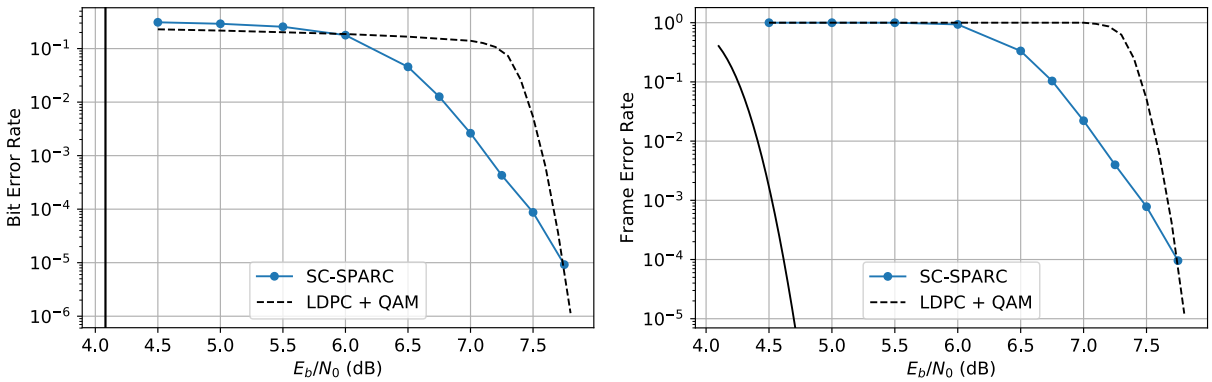


Figure 2.6: Error performance of complex SC-SPARCs defined via an $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix. Code parameters: $R = 1.6$ bits/dimension, $L = 960$, $M = 128$, code length $n = 2100$. The dashed lines show the performance of coded modulation: $(K = 6480, N = 16200)$ DVB-S2 LDPC + 256 QAM, frame length = 2025 symbols. The solid black line in the BER plot is the AWGN Shannon limit for $R = 1.6$ bits/dimension, and in the FER plot, it is the normal approximation to the AWGN finite length error probability bound in [34].

Fig. 2.6 shows the performance of an SC-SPARC with a shorter code length $n = 2100$, and a rate of 1.6 bits/dimension. The AMP decoder for the SC-SPARC is run for a maximum of 100 iterations. The coded modulation scheme uses a rate $\frac{1}{2}$ (6480, 16200) DVB-S2 LDPC code with 256-QAM modulation, for the same overall rate of 1.6 bits/dimension and a frame length of 2025 symbols. We observe that the SC-SPARC requires a smaller E_b/N_0 to achieve BERs in the range 10^{-1} to 10^{-5} and FERs down to 10^{-4} compared to the coded modulation scheme. However, for BERs and FERs lower than 10^{-5} and 10^{-4}, respectively, we expect the coded modulation scheme to require a smaller E_b/N_0 because its error rate drops faster as E_b/N_0 increases.

In Fig. 2.7, the rate of the SC-SPARC is 2 bits/dimension and the code length is $n = 2688$. The AMP decoder for the SC-SPARC is run for a maximum of 100 iterations. The coded

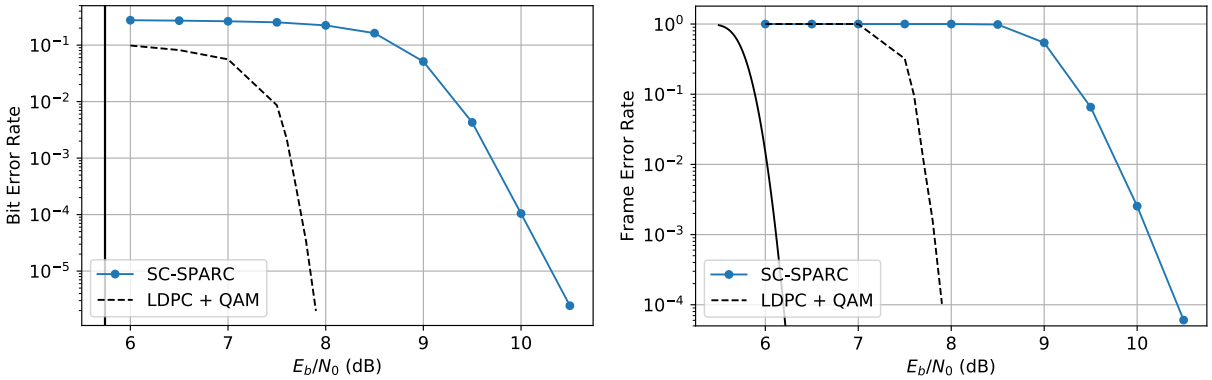


Figure 2.7: Error performance of complex SC-SPARCs defined via an $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix. Code parameters: $R = 2$ bits/dimension, $L = 2688$, $M = 16$, code length $n = 2688$. The dashed lines show the performance of coded modulation: $(K = 10800, N = 16200)$ DVB-S2 LDPC + 64 QAM, frame length = 2700 symbols. The solid black line in the BER plot is the AWGN Shannon limit for $R = 2$ bits/dimension, and in the FER plot, it is the normal approximation to the AWGN finite length error probability bound in [34].

modulation scheme uses a rate $\frac{2}{3}$ (10800, 16200) DVB-S2 LDPC code with 64-QAM modulation, for the same overall rate of 2 bits/dimension and a frame length of 2700 symbols. We observe that the SC-SPARC has a higher BER and FER compared to the coded modulation scheme for all values of E_b/N_0, and its error rate also drops more slowly as E_b/N_0 increases.

In the above plots, the SC-SPARC parameters $(\omega, \Lambda, L, M, n)$ have not been carefully optimised. An interesting direction for future work is to develop good finite length design guidelines for choosing these parameters as a function of rate and snr. Another direction is to explore whether alternative base matrix designs could improve the finite length performance at higher rates like 2 bits/dimension.

2.5.1 Implementation details

The $(\omega, \Lambda, \rho = 0)$ base matrix was used for all the simulations. Furthermore, to reduce the decoding complexity and the memory requirement, a few modifications were made to the SC-SPARC construction and the AMP decoder.

DFT based design matrices We replaced the Gaussian (or complex Gaussian) design matrix with either a Hadamard or discrete Fourier transform (DFT) based design matrix. This enables the matrix-vector multiplications in the AMP decoder (2.7)–(2.8) to be computed via the fast Walsh-Hadamard transform (FWHT) [59] or the fast Fourier transform (FFT) [60], which significantly lowers the decoding complexity and memory requirement. Our approach follows that of [22, 56] where Hadamard based design matrices were used for real-valued SPARCs.

A Hadamard-based design matrix $\boldsymbol{A}$ can be generated as follows given base matrix $\boldsymbol{W} \in \mathbb{R}_+^{\mathsf{R}\times\mathsf{C}}$. Let $k = \lceil \log_2(\max((n/\mathsf{R}) + 1, (LM/\mathsf{C}) + 1)) \rceil$. Each *block* of the design matrix (indexed by (r, c) for $\mathsf{r} \in [\mathsf{R}], \mathsf{c} \in [\mathsf{C}]$) is constructed by choosing (n/R) rows and (LM/C) columns

uniformly at random[3] from a $2^k \times 2^k$ Hadamard matrix and then scaling each entry up by $\sqrt{W_{\mathsf{rc}}/L}$. The resulting design matrix has entries $A_{ij} = \pm\sqrt{W_{\mathsf{r}(i),\mathsf{c}(j)}/L}$. When DFT based design matrices are used, the FFT operation must be appropriately normalised.[4]

The computational complexity of the AMP decoder is dominated by the two matrix-vector multiplications associated with the design matrix $\boldsymbol{A}$. These operations have complexity $O(nLM)$ when $\boldsymbol{A}$ has independent Gaussian entries. The memory requirements of the encoder and decoder are also proportional to nLM since the Gaussian design matrix has to be stored. By constructing the design matrix using randomly sampled rows of the (deterministic) Hadamard or DFT matrix, the complexity of the matrix-vector multiplications (replaced by FWHTs/FFTs) is reduced to $O(LM\log(LM))$, and the memory requirements of the encoder and decoder are proportional to ωLM. The error performance of Hadamard/DFT based design matrices was found to be similar to that of Gaussian matrices for large matrix sizes.

Online estimation of state evolution parameters The AMP decoder in (2.7)–(2.8) contains parameters computed using the state evolution (SE) recursion (2.9)–(2.11). In particular, the vector $\widetilde{\boldsymbol{v}}^t$, and the matrix $\widetilde{\boldsymbol{Q}}^t$ are determined via SE parameters computed offline. Instead of computing the SE parameters offline, the SE parameters can be estimated online (at runtime) using the outputs of the AMP decoder in each iteration. The SE parameters $\{\gamma_{\mathsf{r}}^t\}_{\mathsf{r}\in[\mathsf{R}]}$, $\{\phi_{\mathsf{r}}^t\}_{\mathsf{r}\in[\mathsf{R}]}$ and $\{\tau_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$, which are needed to compute $\widetilde{\boldsymbol{v}}^t$ and $\widetilde{\boldsymbol{Q}}^t$ (see (2.12)) can be estimated online in the following way. For $\mathsf{r}\in[\mathsf{R}]$ and $\mathsf{c}\in[\mathsf{C}]$,

$$\hat{\gamma}_{\mathsf{r}}^t = \frac{1}{\mathsf{C}}\sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}}\bigg(1 - \frac{\|\boldsymbol{\beta}_{\mathsf{c}}^t\|^2}{L/\mathsf{C}}\bigg), \tag{2.50}$$

$$\hat{\phi}_{\mathsf{r}}^t = \begin{cases} \sigma^2 + \hat{\gamma}_{\mathsf{r}}^t & \text{if the decoder knows } \sigma^2, \\ \frac{\|\boldsymbol{z}_{\mathsf{r}}^t\|^2}{n/\mathsf{R}} & \text{otherwise,} \end{cases} \tag{2.51}$$

$$\hat{\tau}_{\mathsf{c}}^t = \frac{L}{n}\left[\frac{1}{\mathsf{R}}\sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\hat{\phi}_{\mathsf{r}}^t}\right]^{-1}. \tag{2.52}$$

The justification for these estimates comes from [1, Lemma 7.6], which proves that the estimates $\hat{\gamma}_{\mathsf{r}}^t, \hat{\phi}_{\mathsf{r}}^t$ concentrate on $\gamma_{\mathsf{r}}^t, \phi_{\mathsf{r}}^t$, respectively, for large (n, L). We observe that using online estimates of the SE parameters results in better empirical error performance than using deterministic SE parameters. A similar improvement was observed in [47] for power allocated SPARCs.

[3] Earlier works [22, 47] considered randomly choosing the rows of the Hadamard matrix, but not the columns. In our simulations we found that at smaller design matrix sizes, the AMP decoder sometimes diverged when design matrices were based on randomly chosen rows only. Randomly choosing both the rows and the columns led to improved performance.

[4] We do not use the first row and column of the Hadamard matrix because they are all +1's. The other rows and columns have an equal number of +1's and −1's. We do not use the first and $(2^{k-1}+1)$-th row and column of the DFT matrix because their entries are all real-valued.

Early stopping of AMP Since the online estimates of SE parameters in (2.50)–(2.52) are estimates of certain noise variances related to the decoding error in each iteration of the AMP, we choose to stop the AMP decoder early if the change in $\hat{\boldsymbol{\gamma}}^t, \hat{\boldsymbol{\phi}}^t$ or $\hat{\boldsymbol{\tau}}^t$ falls below a prescribed threshold over consecutive iterations. A similar stopping criterion was used in [47] to terminate the AMP decoder for power allocated SPARCs.

A Python implementation of SPARCs (both power allocated and spatially coupled) with AMP decoding is available at [155].

2.6 Sliding window AMP decoding

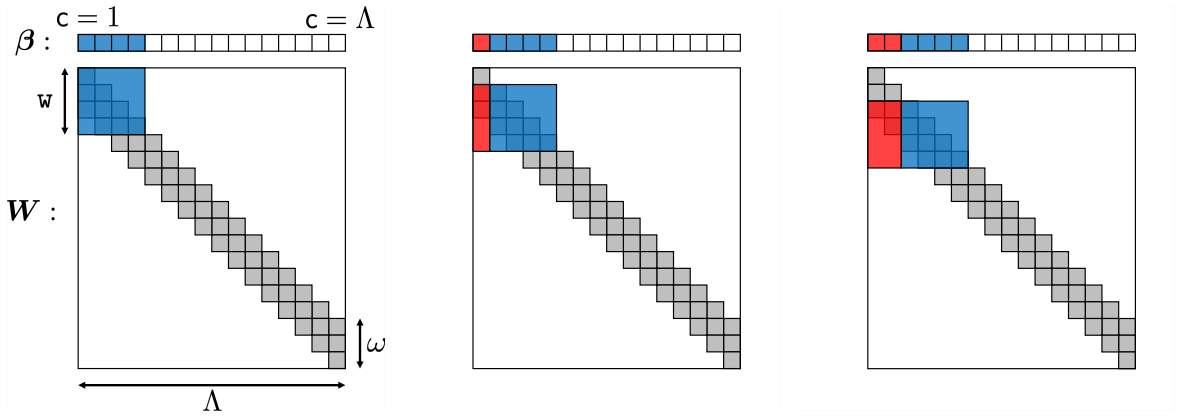
As shown in Fig. 2.2, the decoding progression of SC-SPARCs (at high rates) starts from the two ends of the message vector and progresses towards the centre until the whole message vector is decoded. This decoding progression suggests that the estimates of the sections at the middle of the message vector are not being improved by the decoder during the initial iterations of decoding, and the estimates of the sections at the two ends are not further improved during the final iterations of decoding. Therefore, at different stages of decoding, the decoder is wasting computational power trying to improve the estimates of sections of the message vector where improvements cannot be made.

It is therefore desirable to design a decoder that tracks the decoding progression, updating only the estimates of the sections of the message vector where they can be improved. The computational complexity and memory requirements of such a decoder would depend on the number of sections whose estimate can be improved, which may be much lower than the total number of sections. Moreover, if this decoder only tracks the decoding progression that goes from the front to the back of the message vector (unidirectional), then this decoder can start decoding once it has received enough codeword symbols from the channel to update the estimate of the first section of the message vector. This has potential to greatly reduce latency, and the continuous decoding process (similar to that of convolutional codes) may be useful in streaming applications.

The same "wave-like" decoding progression is seen in spatially coupled LDPC (SC-LDPC) codes with belief propagation (BP) decoding. A *window decoder* for SC-LDPC codes was proposed and analysed in [156, 157]. The sliding window AMP decoder we describe below borrows many ideas from [156].

2.6.1 Sliding window decoder description

Recall that a SC-SPARC is defined by a design matrix $\boldsymbol{A} \in \mathbb{R}^{n \times LM}$ that is constructed using a base matrix $\boldsymbol{W} \in \mathbb{R}_+^{\mathsf{R} \times \mathsf{C}}$. The codewords are constructed according to $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^{LM}$ is a message vector with L sections. The design matrix $\boldsymbol{A}$ is divided into $\mathsf{R} \times \mathsf{C}$ blocks, and the message vector can be divided into C blocks corresponding to the C column-blocks of



(a) window position 1 (first) (b) window position 2 (c) window position 3

(d) window position 13 (e) window position 14 (f) window position 15 (last)

Figure 2.8: Sliding window decoding steps for window width $\mathtt{w} = 4$ and base matrix parameters ($\omega = 3$, $\Lambda = 16$, $\rho = 0$). Blue: the decoding window and the corresponding parts of the message vector estimate being updated. Red: additional columns on the left of decoding window required by the decoder and the corresponding parts of the message vector (already decoded). Green: parts of the message vector that have been decoded and are no longer used by the decoder.

$\boldsymbol{A}$. We describe the sliding window AMP decoder for $(\omega, \Lambda, \rho = 0)$ base matrices which have $\mathsf{R} = \Lambda + \omega - 1$ rows, $\mathsf{C} = \Lambda$ columns, and ω non-zero entries in each column (see Def. 2.1.1). The same ideas can be extended to other spatially coupled base matrix designs that have a band diagonal structure.

The progression of the sliding window decoder is illustrated in Fig. 2.8. The high level idea is as follows. A square $\mathtt{w} \times \mathtt{w}$ *decoding window* (blue) moves diagonally across the base matrix $\boldsymbol{W}$, from top-left to bottom-right, to eventually cover all the non-zero entries of the base matrix in its progression. At each *window position*, the decoder acts on the part of the noisy channel output $\boldsymbol{y}$ that corresponds to the rows of the base matrix within the window, and uses them to estimate of the part of the message vector $\boldsymbol{\beta}$ that corresponds to the columns of the base matrix within the window. We now describe the sliding window decoder in more detail.

The sliding window AMP decoder has an additional *window width* parameter $\mathtt{w}$ compared to the "full" AMP decoder in (2.7)–(2.8). This parameter determines the size of the decoding

window and lies within the range ω to R. When $\mathtt{w} = \mathsf{R}$ the window decoder is the same as the full decoder.

On initialisation (Fig. 2.8a), the initial estimate of the message vector is set to the all zero vector, and the decoding window (blue) covers the top-left $\mathtt{w}$-by-$\mathtt{w}$ entries of the base matrix. The AMP decoder (2.7)–(2.8) is run for this window with the corresponding parts of the channel output and message vector. It is as if the AMP decoder is decoding an SC-SPARC constructed using a base matrix equal to the "sub-matrix" covered by the decoding window. The estimates of the message vector in this window (blue) are updated iteratively. The decoder runs until the change in the estimates corresponding to the left-most column in the decoding window — the target column — drops below a tolerance parameter (or when the maximum number of iterations allowed per window position is reached).

The decoding window (blue) then moves to the next *window position* by shifting one row down and one column to the right in the base matrix (Fig. 2.8b). The estimate of the message vector corresponding to the previous target column ($\mathtt{c} = 1$) is now fixed and will not be updated further. (Parts of the message vector that are no longer being updated are shown in either red or green.) The AMP decoder is run again for this new decoding window, until the change in the estimates of the message vector corresponding to the new target column ($\mathtt{c} = 2$) drops below the tolerance parameter. This process of "shift and decode" continues until the decoding window covers the last row of the base matrix, i.e., the target column position reaches $\mathtt{c} = \mathsf{R} - \mathtt{w} + 1 = \Lambda + \omega - \mathtt{w}$. At this position the AMP decoder runs until the change in the estimates of the message vector corresponding to *all* the columns in the decoding window drops below the tolerance parameter. Then the sliding window decoder terminates.

After the first window position (Fig. 2.8a), the AMP decoder requires up to $\omega - 1$ additional columns (red) on the left of the decoding window (blue) in order to correctly calculate the residual term $\boldsymbol{z}^t$ in (2.7). Note that the estimates of the message vector corresponding to those additional columns are fixed, hence those additional columns are not needed in the calculation of $\boldsymbol{\beta}^{t+1}$ in (2.8).

As the sliding window AMP decoder moves across the base matrix, it outputs soft estimates for parts of the message vector. During this process, the parts of the message vector that have been decoded and are no longer used by the decoder (green) can be hard decoded (set largest entry in each section to 1 and the remaining entries to 0).

Decoding latency and complexity

In practice, the sliding window AMP decoder offers a way to trade-off between computational complexity, latency and error performance. Indeed, the full decoder is recovered when the window width $\mathtt{w}$ equals $\mathsf{R} = \Lambda + \omega - 1$.

The decoding latency of the window decoder in number of received symbols is $\mathtt{w} \cdot \frac{n}{\Lambda+\omega-1}$, compared to n for the full decoder. When a Hadamard or DFT based design matrix is used

together with their corresponding fast transform (see Sec. 2.5.1), the per-iteration decoding complexity of the window decoder is smaller than that of the full decoder by a factor of approximately Λ/ω. Specifically, the per-iteration decoding complexity of the window decoder is $O(\mathtt{w}\,\omega \cdot x \log x)$, where $\mathtt{w}\,\omega$ is the maximum number of non-zero entries in the base matrix covered by the decoding window, and $x = LM/\Lambda$ is the number of columns in each column block of the design matrix. Observe that both the decoding latency and complexity scale linearly with the window width $\mathtt{w}$. Note that it reasonable to consider L and n scaling linearly with Λ.

We expect to see significant complexity and latency advantages of the window decoder over the full decoder for long code lengths because the complexity and latency of the window decoder depends on the window width instead of the code length (assuming fixed ω and L/Λ).

Asymptotic state evolution analysis

Using the asymptotic (section size $M \to \infty$) state evolution analysis in Section 2.3.1, one can see that the (unidirectional) sliding window AMP decoder with window width $\mathtt{w} = \omega$ has the same asymptotic error performance as the full decoder. It also requires approximately twice the number of iterations to finish decoding. However, this assumes that the sliding window decoder knows exactly how many columns to shift between window positions so that all decoded sections of the message vector are no longer within the decoding window at the next window position. The sliding window decoder described in the previous section only shifts one column between window positions.

Disadvantages compared to the full AMP decoder

1. The sliding window decoder cannot utilise the "seed" at the end (bottom-right) of the base matrix until the last few window positions. Therefore, it has worse error performance compared to the full decoder.

2. The sliding window decoder does not track the decoding progression ("wave") from both ends of the message vector simultaneously. Therefore, it takes more iterations to decode.

3. If decoding errors occur early on in sliding window decoding, they propagate to the subsequent window positions and cannot be corrected since the decoding window never moves in the reverse direction. This can lead to decoding instances where many section errors occur.

The first two disadvantages can be addressed by using a *bidirectional* sliding window decoder. However, bidirectional window decoders loose the low latency advantage of unidirectional window decoders, since the whole codeword has to be received before decoding can initiate.

The window extension algorithm proposed in [158] mitigates the effects of error propagation in the window decoding of SC-LDPC codes. We expect a similar approach to help address the

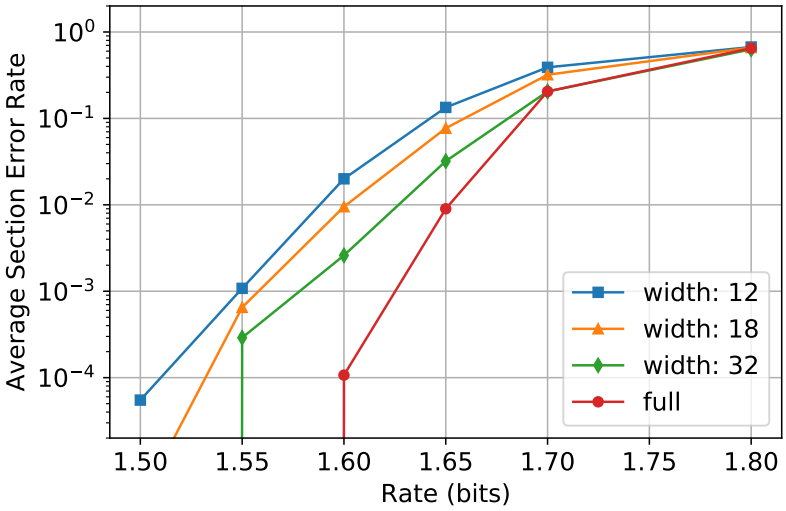


Figure 2.9: Average SER of SC-SPARCs with sliding window AMP decoding as the window width `w` increases. Code parameters: $C = 2$ bits, $\omega = 6$, $\Lambda = 64$, $\rho = 0$, $M = 512$ and $L = 2048$. The vertical lines indicate that no section errors were observed over 10^4 trials at smaller rates.

third disadvantage of sliding window AMP decoding of SC-SPARCs.

2.6.2 Empirical performance

In this section, we investigate the finite length error performance of sliding window AMP decoding via numerical simulations. As in Section 2.5, we Hadamard based design matrices in these simulations for lower complexity and memory requirements.

Fixed code length, varying window width Fig. 2.9 shows the average section error rate (SER) of the sliding window decoder for different values of the window width `w`, with the other code parameters fixed. The SC-SPARCs are constructed using an $(\omega = 6, \Lambda = 64, \rho = 0)$ base matrix and has code parameters $M = 512$ and $L = 2048$. We observe that the error performance improves as the window width is increased. Recall that the decoding latency and per iteration complexity increases with the window width.

According to the asymptotic state evolution recursion, the sliding window decoder with window width $\mathtt{w} = \omega$ achieves the same asymptotic performance as full decoder. However, at finite code lengths n and section sizes M, the column blocks of the message vector are not perfectly decoded (i.e., corresponding MSE $\neq 0$) at the end of the predicted iteration using $\mathtt{w} = \omega$. Therefore, larger window widths improve the error performance by using larger decoding windows (more channel output information can be used), and updating column blocks of message vector estimates for a larger number of iterations (due to being in more window positions). Moreover, we increase the number of decoding iterations allowed per window position for larger window widths, as the window decoder requires more iterations for the estimate of the target column to converge. See Table 2.1 for the list of decoding iterations allowed for different window

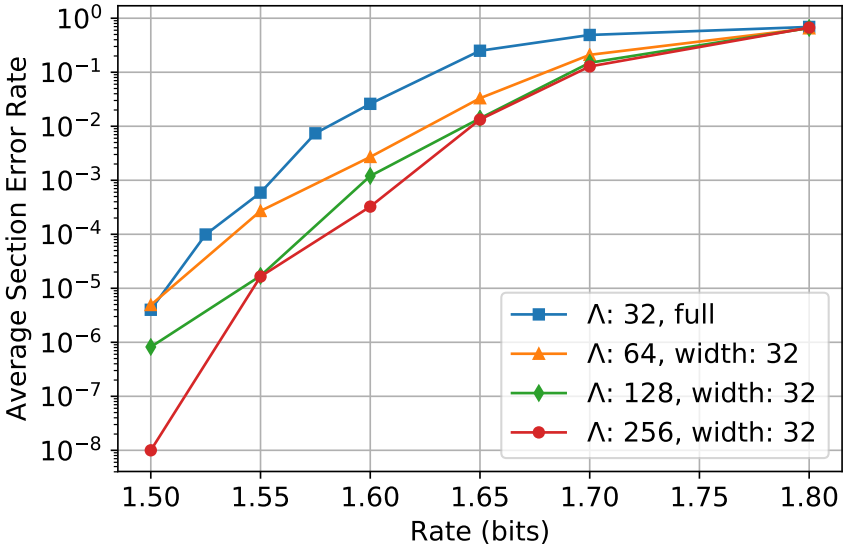


Figure 2.10: Average SER of SC-SPARCs with sliding window AMP decoding as the code length increases and the window width is fixed to $\mathtt{w} = 32$. Code parameters: $C = 2$ bits, $\omega = 6$, $\rho = 0$, $M = 512$, and $L/\Lambda = 32$ sections per column block. ($\Lambda = 32$ corresponds to full AMP decoding.)

Table 2.1: Maximum number of iterations allowed (per window position for the window decoder)

	full decoder ($\omega = 6$)		window decoder			
	$\Lambda = 32$	$\Lambda = 64$	$\mathtt{w} = 6$	$\mathtt{w} = 12$	$\mathtt{w} = 18$	$\mathtt{w} = 32$
$t_{\max}$	100	200	10	20	30	100

widths.

Fixed window width, increasing code length Fig. 2.10 shows the performance of the sliding window AMP decoder with fixed window width $\mathtt{w} = 32$, and increasing code length. The SC-SPARCs are constructed using $(\omega, \Lambda, \rho = 0)$ base matrices with a fixed coupling width $\omega = 6$ and has section size $M = 512$. The number of sections L (and code length) increases proportionally with Λ, with $L/\Lambda = 32$ held constant. Since the window width $\mathtt{w}$, coupling width ω, section size M and L/Λ are fixed, the per iteration decoding complexity is the same for the four different cases in Fig. 2.10. All decoders in this set of simulations have the maximum number of iterations allowed (per window position) set to 100. Note that for the $\Lambda = 32$ case, the full AMP decoder is used.

There is a trade-off in error performance when the coupling length Λ is increased whilst the coupling width ω and decoding window width $\mathtt{w}$ are held constant. Increasing Λ with ω fixed reduces the rate loss of the SC-SPARC (2.5). This results in a lower inner SPARC rate in the SC-SPARC, which leads to better error performance when its decoded with the full decoder. However, window decoders with a fixed window width become more suboptimal compared to full decoders as the coupling length Λ increases. In Fig. 2.10 we observe that the section error rate generally improves as Λ (and hence the code length) is increased, for a fixed window

width $\mathtt{w}$. For large values of Λ such as $\Lambda = 128$ or 256, the complexity of full AMP decoding is prohibitively large. In such cases, sliding window AMP decoding (green and red) offers a noticeable improvement over smaller code length SC-SPARCs with full AMP decoding (blue).

2.7 Proof of Proposition 2.3.1

Using the definition in (2.4), the state evolution equations (2.9)–(2.10) for the (ω, Λ, ρ) base matrix are as follows. With $\psi_{\mathsf{c}}^0 = 1$ for $\mathsf{c} \in [\Lambda]$, for $t \geq 0$:

$$\phi_{\mathsf{r}}^t = \sigma^2 \left[1 + \vartheta \, \mathsf{snr} \left(\frac{1-\rho}{\omega} \sum_{\mathsf{c}=\underline{\mathsf{c}}_{\mathsf{r}}}^{\bar{\mathsf{c}}_{\mathsf{r}}} \psi_{\mathsf{c}}^t + \frac{\rho}{\Lambda - 1} \sum_{\mathsf{c} \in [\Lambda] \setminus \{\underline{\mathsf{c}}_{\mathsf{r}}, \ldots, \bar{\mathsf{c}}_{\mathsf{r}}\}} \psi_{\mathsf{c}}^t \right) \right], \quad \mathsf{r} \in [\Lambda + \omega - 1], \tag{2.53}$$

$$\nu_{\mathsf{c}}^t = \frac{R}{\tau_{\mathsf{c}}^t \ln M} = \frac{(1-\rho)\, P}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{\mathsf{c}+\omega-1} \frac{1}{\phi_{\mathsf{r}}^t} + \frac{\rho \, P}{\Lambda - 1} \sum_{\mathsf{r} \in [\Lambda+\omega-1] \setminus \{\mathsf{c}, \ldots, \mathsf{c}+\omega-1\}} \frac{1}{\phi_{\mathsf{r}}^t}, \quad \mathsf{c} \in [\Lambda] \tag{2.54}$$

$$\psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(R/(\nu_{\mathsf{c}}^t \ln M)). \tag{2.55}$$

Here $\mathcal{E}(\cdot)$ is defined in (2.11), and $\underline{\mathsf{c}}_{\mathsf{r}}, \bar{\mathsf{c}}_{\mathsf{r}}$ are defined (2.27).

Since the variables ψ_{c}^t for $\mathsf{c} \in [\Lambda]$ and $t \geq 0$ are symmetric about the center column index, i.e. $\psi_{\mathsf{c}}^t = \psi_{\Lambda-\mathsf{c}+1}^t$ for $\mathsf{c} \leq \lceil \frac{\Lambda}{2} \rceil$, we carry out the analysis for $\mathsf{c} \leq \lceil \frac{\Lambda}{2} \rceil$; the result for the other half then holds by symmetry. We will upper bound ψ_{c}^t using Lemma 2.3.1. Using (2.21), for the first iteration we will have $\psi_{\mathsf{c}}^1 \leq f_{M,\delta}$ for indices c for which $\{\nu_{\mathsf{c}}^0 > (2+\delta)R\}$. We now obtain a lower bound on ν_{c}^0 for indices $\mathsf{c} < \omega$.

Using (2.54) we have

$$\begin{aligned}
\nu_{\mathsf{c}}^0 &= \frac{(1-\rho)P}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{\mathsf{c}+\omega-1} \frac{1}{\phi_{\mathsf{r}}^0} + \frac{\rho P}{\Lambda - 1} \sum_{\mathsf{r} \in [\Lambda+\omega-1] \setminus \{\mathsf{c}, \ldots, \mathsf{c}+\omega-1\}} \frac{1}{\phi_{\mathsf{r}}^0} \qquad (2.56) \\
&\geq \frac{(1-\rho)P}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{\mathsf{c}+\omega-1} \frac{1}{\phi_{\mathsf{r}}^0} \\
&\overset{\text{(i)}}{=} \frac{(1-\rho)\,\mathsf{snr}}{\omega} \left(\sum_{\mathsf{r}=\mathsf{c}}^{\omega-1} \frac{1}{1 + \frac{(1-\rho)\,\vartheta\,\mathsf{snr}}{\omega} \mathsf{r} + \frac{\rho\,\vartheta\,\mathsf{snr}}{\Lambda-1}(\Lambda - \mathsf{r})} + \frac{\mathsf{c}}{1 + (1-\rho)\,\vartheta\,\mathsf{snr} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\omega}{\Lambda-1}} \right) \\
&\geq \frac{(1-\rho)\,\mathsf{snr}}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{\omega-1} \frac{1}{1 + \rho\,\vartheta\,\mathsf{snr} + (1-\rho)\,\vartheta\,\mathsf{snr}\,\frac{\mathsf{r}}{\omega}} + \frac{\mathsf{c}}{\omega} \frac{(1-\rho)\,\mathsf{snr}}{1 + \vartheta\,\mathsf{snr}} \\
&\overset{\text{(ii)}}{\geq} \frac{1}{\vartheta} \left(\ln(1 + \vartheta\,\mathsf{snr}) - \ln\left(1 + \rho\,\vartheta\,\mathsf{snr} + (1-\rho)\,\vartheta\,\mathsf{snr}\,\frac{\mathsf{c}}{\omega}\right) \right) + \frac{\mathsf{c}}{\omega} \frac{(1-\rho)\,\mathsf{snr}}{1 + \vartheta\,\mathsf{snr}} \\
&\overset{\text{(iii)}}{\geq} \frac{1}{\vartheta} \ln(1 + \vartheta\,\mathsf{snr}) - \rho\,\mathsf{snr} - (1-\rho)\,\mathsf{snr}\,\frac{\mathsf{c}}{\omega} + \frac{\mathsf{c}}{\omega} \frac{(1-\rho)\,\mathsf{snr}}{1 + \vartheta\,\mathsf{snr}} \\
&\geq \frac{1}{\vartheta} \ln(1 + \vartheta\,\mathsf{snr}) - \rho\,\mathsf{snr} - \frac{\mathsf{c}}{\omega} \frac{\vartheta\,\mathsf{snr}^2}{1 + \vartheta\,\mathsf{snr}},
\end{aligned}$$

where the labelled steps are obtained as follows: (i) using the expression for ϕ_{r}^t in (2.53) and the

fact that $\mathsf{c} < \omega$, (ii) using a definite integral to lower bound the left Riemann sum of a positive decreasing function:

$$\frac{1}{\omega}\sum_{\mathsf{r}=\mathsf{c}}^{\omega-1}\frac{1}{\frac{1+\rho\,\vartheta\,\mathsf{snr}}{(1-\rho)\,\vartheta\,\mathsf{snr}}+\frac{\mathsf{r}}{\omega}} \geq \int_{\mathsf{c}/\omega}^{1}\frac{1}{\frac{1+\rho\,\vartheta\,\mathsf{snr}}{(1-\rho)\,\vartheta\,\mathsf{snr}}+x}\,\mathrm{d}x. \tag{2.57}$$

Inequality (iii) is obtained using $\ln(1+x) \leq x$.

Therefore, the condition $\nu_{\mathsf{c}}^0 > (2+\delta)R$ will be satisfied if

$$\frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \rho\,\mathsf{snr} - \frac{\mathsf{c}}{\omega}\frac{\vartheta\,\mathsf{snr}^2}{1+\vartheta\,\mathsf{snr}} \;>\; (2+\delta)R. \tag{2.58}$$

Rearranging (2.58) gives

$$\mathsf{c} \;<\; \omega\cdot\frac{2(1+\vartheta\,\mathsf{snr})}{\vartheta\,\mathsf{snr}^2}\left[\frac{1}{2\vartheta}\ln\left(1+\vartheta\,\mathsf{snr}\right) - \frac{\rho\,\mathsf{snr}}{2} - R - \frac{\delta R}{2}\right], \tag{2.59}$$

Note that the RHS of (2.59) is smaller than or equal to 1 if $R \geq \frac{(1-\rho)\mathsf{snr}}{(2+\delta)(1+\vartheta\,\mathsf{snr})}$. Using $\rho \leq \frac{\Delta}{3\mathsf{snr}}$ and $\delta < \frac{\Delta}{2R}$, the sufficient condition in (2.59) for $\psi_{\mathsf{c}}^1 \leq f_{M,\delta}$ can be weakened to $\mathsf{c} \leq g$ where g is defined in (2.42). Note that the condition (2.40) on ω guarantees that $g > 1$.

Notice from (2.56) that ν_{c}^0 is decreasing in c for $\mathsf{c} \in [1,\omega]$ and is then constant for $\mathsf{c} \in [\,\omega, \lceil\frac{\Lambda}{2}\rceil]$. Therefore, for any $\delta \in (0,1)$, if $\nu_{\mathsf{c}}^0 > (2+\delta)R$ is satisfied for $\mathsf{c} = \omega$ then $\psi_{\mathsf{c}}^1 \leq f_{M,\delta}$ for all $\mathsf{c} \in [\Lambda]$. By using a similar analysis as above for lower bounding ν_{c}^0, one can show that a sufficient condition for $\nu_{\omega}^0 > (2+\delta)R$ is $R < \frac{(1-\rho)\mathsf{snr}}{(2+\delta)(1+\vartheta\,\mathsf{snr})}$.

Next we consider subsequent iterations $t > 1$. Assume towards induction that

$$\psi_{\mathsf{c}}^t \leq f_{M,\delta}, \quad \text{for } \mathsf{c} \leq g_t, \tag{2.60}$$

where $g_t \geq tg$. We will prove that (2.60) implies $\psi_{\mathsf{c}}^{t+1} \leq f_{M,\delta}$ for $\mathsf{c} \leq g_t + g$. We prove the result for $g_t \geq \omega$, with the other case being similar. We wish to find column indices $\mathsf{c} \in (g_t, g_t+\omega)$ for which $\psi_{\mathsf{c}}^{t+1} \leq f_{M,\delta}$, or equivalently $\nu_{\mathsf{c}}^t > (2+\delta)R$. For brevity, we will use the shorthand $f := f_{M,\delta}$. Using the induction assumption (2.60) in (2.53), we deduce

$$\frac{\phi_{\mathsf{r}}^t}{\sigma^2} \leq \begin{cases} 1 + f\,(1-\rho)\,\vartheta\,\mathsf{snr}\,\frac{\mathsf{r}}{\omega} + f\,\rho\,\vartheta\,\mathsf{snr}\,\frac{g_t-\mathsf{r}}{\Lambda-1} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-g_t}{\Lambda-1}, & 1 \leq \mathsf{r} \leq \omega, \\ 1 + f\,(1-\rho)\,\vartheta\,\mathsf{snr} + f\,\rho\,\vartheta\,\mathsf{snr}\,\frac{g_t-\omega}{\Lambda-1} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-g_t}{\Lambda-1}, & \omega \leq \mathsf{r} \leq g_t, \\ 1 + \frac{(1-\rho)\,\vartheta\,\mathsf{snr}}{\omega}\left[f(\omega-(\mathsf{r}-g_t)) + (\mathsf{r}-g_t)\right] + f\,\rho\,\vartheta\,\mathsf{snr}\,\frac{\mathsf{r}-\omega}{\Lambda-1} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\mathsf{r}}{\Lambda-1}, & g_t \leq \mathsf{r} < g_t+\omega, \\ 1 + (1-\rho)\,\vartheta\,\mathsf{snr} + f\,\rho\,\vartheta\,\mathsf{snr}\,\frac{\mathsf{r}-\omega}{\Lambda-1} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\mathsf{r}}{\Lambda-1}, & \mathsf{r} \geq g_t+\omega. \end{cases} \tag{2.61}$$

For M sufficiently large (i.e., f sufficiently small), noting that $g_t \geq \omega$ we can simplify (2.61) to

$$\frac{\phi_{\mathsf{r}}^t}{\sigma^2} \leq \begin{cases} 1 + f\,(1-\rho)\,\vartheta\,\mathsf{snr}\,\frac{\mathsf{r}}{\omega} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\mathsf{r}}{\Lambda-1}, & 1 \leq \mathsf{r} \leq \omega, \\ 1 + f\,(1-\rho)\,\vartheta\,\mathsf{snr} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\omega}{\Lambda-1}, & \omega \leq \mathsf{r} \leq g_t, \\ 1 + \frac{(1-\rho)\,\vartheta\,\mathsf{snr}}{\omega}\,[f(\omega-(\mathsf{r}-g_t)) + (\mathsf{r}-g_t)] + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\omega}{\Lambda-1}, & g_t \leq \mathsf{r} < g_t+\omega, \\ 1 + (1-\rho)\,\vartheta\,\mathsf{snr} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\omega}{\Lambda-1}, & \mathsf{r} \geq g_t+\omega. \end{cases} \tag{2.62}$$

We now obtain a lower bound on ν_{c}^t for $g_t < \mathsf{c} < g_t + \omega$. Using (2.54) we have

$$\begin{aligned}
\nu_{\mathsf{c}}^t &\geq \frac{(1-\rho)P}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{\mathsf{c}+\omega-1} \frac{1}{\phi_{\mathsf{r}}^t} \\
&\overset{\text{(i)}}{\geq} \frac{(1-\rho)\mathsf{snr}}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{g_t+\omega-1} \frac{1}{1 + \frac{(1-\rho)\,\vartheta\,\mathsf{snr}}{\omega}\,[f(\omega-(r-g_t)) + (r-g_t)] + \frac{\rho\,\vartheta\,\mathsf{snr}}{\Lambda-1}\,(\Lambda-\omega)} \\
&\qquad\qquad + \frac{\mathsf{c}-g_t}{\omega}\,\frac{(1-\rho)\,\mathsf{snr}}{1 + (1-\rho)\,\vartheta\,\mathsf{snr} + \rho\,\vartheta\,\mathsf{snr}\,\frac{\Lambda-\omega}{\Lambda-1}} \\
&\geq \frac{(1-\rho)\mathsf{snr}}{\omega} \sum_{\mathsf{r}=\mathsf{c}}^{g_t+\omega-1} \frac{1}{1 + \rho\vartheta\mathsf{snr} + f(1-\rho)\vartheta\mathsf{snr} + (1-f)(1-\rho)\vartheta\mathsf{snr}\frac{(\mathsf{r}-g_t)}{\omega}} + \frac{\mathsf{c}-g_t}{\omega}\,\frac{(1-\rho)\mathsf{snr}}{1+\vartheta\mathsf{snr}} \\
&\geq \frac{(1-\rho)\mathsf{snr}}{\omega} \sum_{\mathsf{r}'=\mathsf{c}-g^t}^{\omega-1} \frac{1}{1 + \rho\vartheta\mathsf{snr} + f(1-\rho)\vartheta\mathsf{snr} + (1-f)(1-\rho)\vartheta\mathsf{snr}\,\frac{\mathsf{r}'}{\omega}} + \frac{\mathsf{c}-g_t}{\omega}\,\frac{(1-\rho)\mathsf{snr}}{1+\vartheta\mathsf{snr}} \\
&\overset{\text{(ii)}}{\geq} \frac{1}{(1-f)\vartheta}\left(\ln(1+\vartheta\,\mathsf{snr}) - \ln\left(1 + \rho\,\vartheta\,\mathsf{snr} + f\,(1-\rho)\,\vartheta\,\mathsf{snr} + (1-f)(1-\rho)\,\vartheta\,\mathsf{snr}\,\frac{\mathsf{c}-g_t}{\omega}\right)\right) \\
&\qquad\qquad + \frac{\mathsf{c}-g_t}{\omega}\,\frac{(1-\rho)\,\mathsf{snr}}{1+\vartheta\,\mathsf{snr}} \\
&\overset{\text{(iii)}}{\geq} \frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \rho\,\mathsf{snr} - f\,(1-\rho)\,\mathsf{snr} - (1-\rho)\,\mathsf{snr}\,\frac{\mathsf{c}-g_t}{\omega} + \frac{\mathsf{c}-g_t}{\omega}\,\frac{(1-\rho)\,\mathsf{snr}}{1+\vartheta\,\mathsf{snr}} \\
&\geq \frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \rho\,\mathsf{snr} - f\,\mathsf{snr} - \frac{\mathsf{c}-g_t}{\omega}\,\frac{\vartheta\,\mathsf{snr}^2}{1+\vartheta\,\mathsf{snr}},
\end{aligned} \tag{2.63}$$

where the labelled steps are obtained as follows: (i) using the bounds for ϕ_{r}^t given in (2.62), (ii) using a definite integral to lower bound the left Riemann sum of a decreasing function, similar to (2.57), and (iii) using the inequalities $\ln(1+x) \leq x$ and $\frac{1}{1-f} \geq 1$.

Recall from Lemma 2.3.1 that $\psi_{\mathsf{c}}^{t+1} \leq f_{M,\delta}$ if $\nu_{\mathsf{c}}^t > (2+\delta)R$. From (2.63), this condition will be satisfied if

$$\frac{1}{\vartheta}\ln(1+\vartheta\,\mathsf{snr}) - \rho\,\mathsf{snr} - f\,\mathsf{snr} - \frac{\mathsf{c}-g_t}{\omega}\,\frac{\vartheta\,\mathsf{snr}^2}{1+\vartheta\,\mathsf{snr}} \;>\; (2+\delta)R. \tag{2.64}$$

Rearranging (2.64) gives

$$\mathsf{c} - g_t < \omega \cdot \frac{2(1+\vartheta\,\mathsf{snr})}{\vartheta\,\mathsf{snr}^2}\left[\frac{1}{2\vartheta}\ln\left(1+\vartheta\,\mathsf{snr}\right) - \frac{\rho\,\mathsf{snr}}{2} - \frac{f\,\mathsf{snr}}{2} - R - \frac{\delta R}{2}\right]. \tag{2.65}$$

Note that the RHS of (2.65) is smaller than or equal to 1 if $R \geq \frac{(1-\rho)\mathsf{snr}}{(2+\delta)(1+\vartheta\,\mathsf{snr})}$. Using $\rho \leq \frac{\Delta}{3\mathsf{snr}}$ and $\delta < \frac{\Delta}{2R}$ in (2.65), we obtain that a sufficient condition for $\psi_{\mathsf{c}}^{t+1} \leq f_{M,\delta}$ is

$$\mathsf{c} - g_t < \omega \cdot \frac{2(1+\vartheta\,\mathsf{snr})}{\vartheta\,\mathsf{snr}^2}\left[\frac{7\Delta}{12} - \frac{f\,\mathsf{snr}}{2}\right] \tag{2.66}$$

For M sufficiently large, $f < \Delta/(6\,\mathsf{snr})$. Thus we conclude from (2.66) that $\psi_{\mathsf{c}}^{t+1} \leq f_{M,\delta}$ for $\mathsf{c} - g_t \leq g$, and hence for $\mathsf{c} \leq (t+1)g$ (since $g_t \geq tg$).

Chapter 3

Modulated sparse regression codes

3.1 Introduction

In this chapter we propose *modulated* sparse regression codes (SPARCs), a generalisation of SPARCs, for communication over the complex AWGN channel.

In the complex AWGN channel, the output symbol y is produced from (complex) input symbol x according to $y = x + w$. The noise random variable w is drawn from a zero mean circularly-symmetric complex Gaussian distribution with variance σ^2, which we denote by $w \sim \mathcal{CN}(0, \sigma^2)$. There is an average power constraint P on the channel input: if a codeword $\boldsymbol{x} = x_1, x_2, \ldots, x_n$ is transmitted over n uses of the channel, then

$$\frac{1}{n}\sum_{i=1}^{n} |x_i|^2 \leq P, \tag{3.1}$$

where $|\cdot|$ denotes the modulus of a complex number. The capacity of the channel is

$$C = \ln\left(1 + \frac{P}{\sigma^2}\right) \text{ nats.} \tag{3.2}$$

As in the standard construction of SPARCs (Section 1.3, Fig. 1.6), a modulated SPARC is defined by a design matrix $\boldsymbol{A}$ of dimension $n \times LM$, where n is the code length. The codeword is $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ has one non-zero entry in each of the L sections of size M.

In the standard SPARC construction, the message is indexed by the locations of the non-zero entries in $\boldsymbol{\beta}$, with their values fixed a priori. Since each section encodes $\ln M$ nats and there are L sections, the rate of the SPARC can be expressed as $R = \frac{L \ln M}{n}$ nats.

In a modulated SPARC, information is encoded in both the locations and the values of the non-zero entries of the message vector $\boldsymbol{\beta}$. In particular, we allow each non-zero entry of $\boldsymbol{\beta}$ to take values in a K-ary constellation with equal probability (e.g., K-ary phase shift keying). Therefore, each section encodes $\ln K + \ln M$ nats, and the rate of the modulated SPARC can

be expressed as

$$R = \frac{L \ln(KM)}{n} \text{ nats.} \tag{3.3}$$

A SPARC without modulation can be seen as having modulation factor $K = 1$.

Modulation introduces an extra degree of freedom in the design of SPARCs which can be used to reduce decoding complexity without sacrificing finite-length error performance (Section 3.5). Furthermore, in Chapter 4 we will see how adding modulation to the SPARC design can be useful in coding for many-user Gaussian multiple access channels.

Since we consider communication over the complex AWGN channel, the SPARC codeword $\boldsymbol{A\beta}$ can be complex-valued. Accordingly, the design matrix $\boldsymbol{A}$ is chosen to have independent zero-mean complex Gaussian entries. We refer to SPARCs defined with complex design matrices as *complex* SPARCs. Furthermore, for modulated complex SPARCs, the non-zero entries of $\boldsymbol{\beta}$ take values in a K-ary complex constellation.

The average power constraint in (3.1) implies that the variances of the entries of $\boldsymbol{A}$ and the non-zero values of $\boldsymbol{\beta}$ should be chosen such that $\|\boldsymbol{A\beta}\|^2 \leq nP$ is satisfied with high probability. For example, if the values of the non-zero entries of $\boldsymbol{\beta}$ are all chosen to be 1 (the unmodulated case), the power constraint is satisfied with high probability if the entries of $\boldsymbol{A}$ are chosen i.i.d. $\sim \mathcal{CN}(0, P/L)$. With optimal decoding, such a design has error probability decaying exponentially in the code length for rates $R < C$ [20].[1]

As introduced in Sections 1.3 and 1.5, either power allocation in the non-zero entries of $\boldsymbol{\beta}$ or spatially coupled design matrices $\boldsymbol{A}$ are required for good error performance with low complexity decoders. In particular, both power allocated and spatially coupled unmodulated SPARCs have been to proven to have vanishing error probability in the large system limit with approximate message passing (AMP) decoding, for any $R < C$ [1, 22, 57, 140]. (Here "large system limit" refers to (L, M, n) all tending to infinity such that $nR = L \ln M$.) In this chapter we will extend the proof to modulated complex SPARCs with either power allocation or spatial coupling. Recall that spatially coupled unmodulated SPARCs and their AMP decoder are described in Chapter 2.

Structure of the chapter

In Section 3.2, we introduce modulated SPARCs with K-ary phase shift keying (PSK) constellations and justify our choice of using PSK. We briefly review how the design matrix $\boldsymbol{A}$ can be constructed from a base matrix, and how both power allocation and spatial coupling can be implemented by an appropriate choice of the base matrix.

In Section 3.3, we propose an AMP decoder for complex SPARCs with PSK modulation and describe its state evolution recursion. The state evolution recursion predicts the mean-squared

[1] The result in [20] was proved for a real-valued SPARC over a real AWGN channel, but the result can be extended to the complex case by similar arguments.

error between the true message vector and its estimate in each iteration of AMP decoding. Since SPARCs have only been analysed for real-valued channels (even in the unmodulated case), we briefly discuss the differences between the AMP decoders and state evolution recursions for real and complex SPARCs (both unmodulated).

In Section 3.4, we analyse the error performance of the AMP decoder for modulated complex SPARCs using state evolution. The main technical result (Proposition 3.4.1) gives an upper bound on a key state evolution parameter which predicts the mean squared error of the AMP estimate in each iteration. Using this bound, we show that in the large system limit, the state evolution recursion for complex SPARCs with K-ary PSK modulation is the same for any fixed value of K, including $K = 1$ (unmodulated). We use this result to prove that K-PSK modulated SPARCs with AMP decoding are asymptotically capacity achieving for the complex AWGN channel, with either spatial coupling or exponentially decaying power allocation (Theorems 2 and 3).

In order to show that K-PSK modulated SPARCs with a specified design are capacity achieving in the large system limit, two steps are required:

1. Prove that for any rate $R < C$, state evolution predicts vanishing probability of decoder error in the large system limit.

2. Prove that the error rate of the AMP decoder is accurately tracked by the state evolution parameters for sufficiently large code length.

In Section 3.4, we carry out Step 1. Step 2 was proved in [1, Thms. 1 and 2] for the case of unmodulated real-valued SPARCs (including spatially coupled and power allocated ones), where it was shown that the normalised mean squared error of the AMP decoder concentrates on the state evolution prediction. Step 2 for modulated complex-valued SPARCs can be proved along the same lines. We do not provide detail on the proof as it is a straightforward extension of the analysis in [1]: the proof uses the same induction argument and sequence of steps as [1], with modifications to account for for $\boldsymbol{A}$ and $\boldsymbol{\beta}$ being complex valued. The key conceptual difference between PSK-modulated complex SPARCs and unmodulated real-valued SPARCs is in the state evolution and its analysis, which is discussed in Sections 3.3 and 3.4.

In Section 3.5, we evaluate the finite length error performance of modulated complex SPARCs with AMP decoding via numerical simulations, and compare their performance with coded modulation schemes using LDPC codes from the DVB-S2 standard. With a DFT-based implementation, the per-iteration complexity of the AMP decoder is $\mathcal{O}(LM(K + \log(LM)))$. For $K \ll \log(LM)$, our numerical results demonstrate that modulation allows one to significantly reduce the decoder complexity without sacrificing error performance.

We would like to mention that SPARCs for the real AWGN channel with binary PSK modulation ($K = 2$) and power allocation were discussed in Adam Greig's PhD thesis [159,

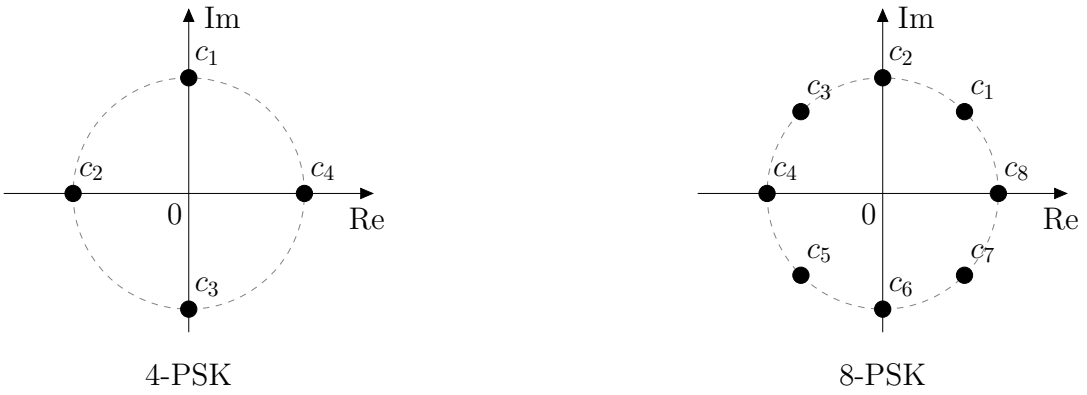


Figure 3.1: Labelling of phase shift keying (PSK) constellation symbols for $K = 4$ and $K = 8$.

Chap. 5]. That work motivated the current study of PSK-modulated SPARCs for the complex AWGN channel.

In this chapter the rate and capacity terms (e.g., R and C) are given in *bits* for the discussion of simulation results (Section 3.5) and given in *nats* elsewhere.

3.2 Modulated SPARC construction

In a modulated SPARC, the value of the non-zero entry in each section of the message vector $\boldsymbol{\beta}$ is chosen from a K-ary constellation. Each section encodes $\ln M$ nats in the location of non-zero entry and at most $\ln K$ nats in the its value, which is achieved when the non-zero values are chosen uniformly from the K-ary constellation (the mapping from bits to constellation symbol is uniform).

In this chapter, we choose the non-zero values in $\boldsymbol{\beta}$ uniformly from a K-ary phase shift keying (PSK) constellation, where the K symbols are equally spaced on the unit circle of the complex plane. The symbols are denoted by

$$c_k \coloneqq e^{\mathrm{j}2\pi k/K} \quad \text{for } k = 1, \ldots, K, \tag{3.4}$$

where $\mathrm{j} \coloneqq \sqrt{-1}$. (Fig. 3.1 illustrates the labelling of the constellation symbols.) PSK is chosen rather than other modulation techniques such as quadrature amplitude modulation (QAM) because the PSK symbols have equal magnitude. Unequal symbol magnitudes will counteract the effect of power allocation and spatial coupling, and simulations show that errors are much more likely to occur in the sections where symbols of smaller magnitude are chosen. We choose the non-zero values uniformly from the PSK constellation due to the global symmetry of the constellation.

3.2.1 Power allocation and spatial coupling

The main idea behind both power allocation and spatial coupling is to change the variances of the (independent Gaussian) entries of $\boldsymbol{A}$ so that certain parts of the message vector are easier to decode than others. Power allocation varies the variance across the sections (columns) of $\boldsymbol{A}$, while spatial coupling varies the variance across both rows and columns such that $\boldsymbol{A}$ has a band diagonal structure. Both techniques can be described in terms of *base matrices*, as described in Chapter 2.1. We now briefly review the base matrix construction and comment on how it can be applied to design matrices with complex Gaussian entries.[2]

The design matrix $\boldsymbol{A} \in \mathbb{C}^{n \times LM}$ is divided into R-by-C equally size *blocks*. The entries within each block are i.i.d. complex Gaussian with zero mean and variance specified by the corresponding entry of a base matrix $\boldsymbol{W} \in \mathbb{R}_+^{\mathsf{R} \times \mathsf{C}}$. The $n \times LM$ design matrix $\boldsymbol{A}$ is constructed by replacing each entry of the base matrix W_{rc}, for $\mathsf{r} \in [\mathsf{R}]$, $\mathsf{c} \in [\mathsf{C}]$, by an $(n/\mathsf{R}) \times (LM/\mathsf{C})$ matrix with entries drawn i.i.d. from $\mathcal{CN}(0, W_{\mathsf{rc}}/L)$. See Fig. 2.1 for an example. Hence, given base matrix $\boldsymbol{W}$, the design matrix $\boldsymbol{A}$ has independent complex Gaussian entries

$$A_{ij} \sim \mathcal{CN}\left(0, \frac{1}{L} W_{\mathsf{r}(i)\mathsf{c}(j)}\right), \quad \text{for } i \in [n],\ j \in [LM]. \tag{3.5}$$

The operators $\mathsf{r}(\cdot) : [n] \to [\mathsf{R}]$ and $\mathsf{c}(\cdot) : [ML] \to [\mathsf{C}]$ in (3.5) map a particular row or column index to its corresponding *row block* or *column block* index. We require C to divide L, resulting in L/C sections per column block.

The non-zero entries of message vector $\boldsymbol{\beta}$ all have magnitude equal to 1 due to PSK modulation. In order to satisfy the average power constraint in (3.1) (in expectation), the entries of the base matrix $\boldsymbol{W}$ must satisfy the condition in (2.2), i.e.,

$$\frac{1}{\mathsf{RC}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}} = P. \tag{3.6}$$

The trivial base matrix with $\mathsf{R} = \mathsf{C} = 1$ corresponds to an uncoupled complex SPARC, i.e., the entries of the design matrix are drawn i.i.d. from $\mathcal{CN}(0, P/L)$. A single-row base matrix with $\mathsf{R} = 1$ and $\mathsf{C} = L$ corresponds to a design matrix with power allocation. For example, the exponential power allocation used in [21, 22, 46] corresponds to

$$W_{1\ell} = LP \cdot \frac{e^{C/L} - 1}{1 - e^{-C}} \cdot e^{-C\ell/L}, \quad \text{for } \ell \in [L], \tag{3.7}$$

where C is the capacity of the complex AWGN channel (in nats) given in (3.2).

[2] In standard unmodulated SPARCs, power allocation is often described as the choice of the non-zero values of the message vector $\boldsymbol{\beta}$ across all L sections while the design matrix $\boldsymbol{A}$ has i.i.d. Gaussian entries. This is equivalent to a choice of the variances of the Gaussian entries across the L sections of the design matrix while the non-zero values of $\boldsymbol{\beta}$ are all equal to 1. The latter perspective allows a unified treatment of power allocation and spatial coupling.

In this chapter we will consider two kinds of base matrices: i) the one corresponding to the exponentially decaying power allocation (3.7), and ii) the class of (ω, Λ, ρ) base matrices used to construct spatially coupled SPARCs defined in Definition 2.1.1.

Recall that an (ω, Λ, ρ) base matrix is described by the coupling width ω, the coupling length Λ, and a parameter $\rho \in [0, 1)$ which specifies the fraction of power allocated to the uncoupled entries in each column. These base matrices have $\mathsf{R} = \Lambda + \omega - 1$ rows, $\mathsf{C} = \Lambda$ columns, with each column having ω identical non-zero entries in the band-diagonal region. When $\rho = 0$, the base matrix only has non-zero entries in the band-diagonal region. For our simulations in Section 3.5, we use $\rho = 0$, whereas for Theorem 2 we choose ρ to be a small positive value proportional to the rate gap from capacity.

In the remainder of the chapter, we use subscripts in sans-serif font (r or c) to denote row or column block indices. For example, $\boldsymbol{\beta}_{\mathsf{c}} \in \mathbb{C}^{LM/\mathsf{C}}$ denotes the c-th column block of $\boldsymbol{\beta} \in \mathbb{C}^{LM}$ for $\mathsf{c} \in [\mathsf{C}]$, and $\boldsymbol{y}_{\mathsf{r}} \in \mathbb{C}^{n/\mathsf{R}}$ denotes the r-th row block of $\boldsymbol{y} \in \mathbb{C}^n$ for $\mathsf{r} \in [\mathsf{R}]$.

3.3 AMP decoder and state evolution

The decoder aims to recover the message vector $\boldsymbol{\beta}$ from the channel output $\boldsymbol{y} \in \mathbb{C}^n$, given by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\beta} + \boldsymbol{w}, \tag{3.8}$$

where $w_1, \ldots, w_n \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, \sigma^2)$. The design matrix $\boldsymbol{A}$ and base matrix $\boldsymbol{W}$ are available to the decoder.

3.3.1 AMP decoder

The approximate message passing (AMP) algorithm is introduced in Section 1.4, and the AMP decoder for unmodulated real-valued SPARCs is introduced in Section 2.2. In this section we propose an AMP decoder for modulated complex SPARCs which takes into account the complex design matrix and the K-ary PSK modulated message vector.[3]

Given the channel output $\boldsymbol{y}$, the AMP decoder iteratively generates message vector estimates $\boldsymbol{\beta}^t$ at iterations $t = 0, 1, 2, \ldots$ as follows: initialises $\boldsymbol{\beta}^0$ to the all-zero vector, and for $t \geq 0$ compute

$$\begin{aligned} \boldsymbol{z}^t &= \boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}^t + \widetilde{\boldsymbol{v}}^t \odot \boldsymbol{z}^{t-1}, \\ \boldsymbol{\beta}^{t+1} &= \eta\big(\boldsymbol{\beta}^t + (\widetilde{\boldsymbol{Q}}^t \odot \boldsymbol{A})^* \boldsymbol{z}^t,\ \widetilde{\boldsymbol{\tau}}^t\big). \end{aligned} \tag{3.9}$$

[3]Complex-valued versions of AMP have been proposed for the linear model in (3.8) where the matrix $\boldsymbol{A}$, as well as $\boldsymbol{\beta}, \boldsymbol{w}$ can be complex [160–162]. However, the complex AMP cannot be directly used to decode SPARCs because it does not incorporate spatial coupling and is based on an i.i.d. prior for $\boldsymbol{\beta}$; in a SPARC the message vector $\boldsymbol{\beta}$ is only section-wise i.i.d.

Here $\odot$ denotes the Hadamard (entry-wise) product, $\boldsymbol{A}^*$ denotes the conjugate transpose of matrix $\boldsymbol{A}$, and quantities with negative time indices are set to zero. The vectors $\widetilde{\boldsymbol{v}}^t \in \mathbb{R}^n$, $\widetilde{\boldsymbol{\tau}}^t \in \mathbb{R}^{LM}$ and the matrix $\widetilde{\boldsymbol{Q}}^t \in \mathbb{R}^{n \times LM}$ are defined in terms of the state evolution parameters which will be described later. The function $\eta = (\eta_1, \ldots, \eta_{LM}) : \mathbb{C}^{LM} \times \mathbb{R}^{LM} \to \mathbb{C}^{LM}$ is defined as follows, for $j \in [LM]$. For j in section $\ell \in [L]$,

$$\eta_j(\boldsymbol{s}, \widetilde{\boldsymbol{\tau}}) = \frac{\sum_{k=1}^{K} c_k \cdot e^{\Re(\overline{s_j} c_k)/\widetilde{\tau}_j}}{\sum_{j' \in \mathrm{sec}(\ell)} \sum_{k'=1}^{K} e^{\Re(\overline{s_{j'}} c_{k'})/\widetilde{\tau}_{j'}}}, \tag{3.10}$$

where $\mathrm{sec}(\ell) \coloneqq \{(\ell-1)M + 1, \ldots, \ell M\}$ denotes the set of indices in section ℓ, and $\{c_k\}_{k \in [K]}$ is the set of PSK symbols defined in (3.4). We use $\overline{z}$ to denote the conjugate of a complex number z, and use $\Re(z)$, $\Im(z)$ to denote its real and imaginary part, respectively. Notice that $\eta_j(\boldsymbol{s}, \widetilde{\boldsymbol{\tau}})$ depends on all the entries of $\boldsymbol{s}$ and $\widetilde{\boldsymbol{\tau}}$ in the section containing j.

Decoder termination When the change in $\boldsymbol{\beta}^t$ across successive iterations falls below a pre-specified tolerance, or the decoder reaches the maximum number of iterations allowed, the decoder terminates. Let T denote the final AMP iteration (in which $\boldsymbol{z}^{T-1}$ and $\boldsymbol{\beta}^T$ are computed). After iteration T, the decoder produces a hard-decision estimate of the message vector, denoted by $\hat{\boldsymbol{\beta}}^T$, as follows. Using $\boldsymbol{s}^{T-1} = \boldsymbol{\beta}^{T-1} + (\widetilde{\boldsymbol{Q}}^{T-1} \odot \boldsymbol{A})^* \boldsymbol{z}^{T-1}$, the ℓ-th section of $\hat{\boldsymbol{\beta}}^T$ is computed as

$$\hat{\boldsymbol{\beta}}^T_{\mathrm{sec}(\ell)} = \underset{\boldsymbol{b} \in \mathcal{B}_{M,K}}{\arg\max}\ \Re\left[\left(\boldsymbol{s}^{T-1}_{\mathrm{sec}(\ell)}\right)^* \boldsymbol{b}\right], \quad \text{for } \ell \in [L], \tag{3.11}$$

where $\mathcal{B}_{M,K}$ denotes the set of all possible length M vectors with a single non-zero entry whose value belongs to the PSK constellation $\{c_k\}_{k \in [K]}$.

Interpretation of the AMP decoder Consider the function $\eta(\cdot, \widetilde{\boldsymbol{\tau}}^t)$ in (3.9), which produces the updated message vector estimate $\boldsymbol{\beta}^{t+1}$. The first input to this function, denoted by $\boldsymbol{s}^t$, can be viewed as a noisy version of the true message vector $\boldsymbol{\beta}$. In particular, $\boldsymbol{s}^t$ is approximately distributed as $\boldsymbol{\beta} + \sqrt{\widetilde{\boldsymbol{\tau}}^t} \odot \boldsymbol{u}$, with $\boldsymbol{u} = [u_1, \ldots, u_{LM}]^T \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, 2)$ independent of $\boldsymbol{\beta}$. Under the above distributional assumption, the function η_j defined in (3.10) is the minimum mean squared error (MMSE) estimator of β_j. That is, for $j \in [LM]$,

$$\eta_j(\boldsymbol{s}, \widetilde{\boldsymbol{\tau}}) = \mathbb{E}\Big[\beta_j \,\Big|\, \boldsymbol{s} = \boldsymbol{\beta} + \sqrt{\widetilde{\boldsymbol{\tau}}} \odot \boldsymbol{u}\Big], \tag{3.12}$$

where the expectation is calculated over $\boldsymbol{\beta}$ and $\boldsymbol{u}$, with the location and value of the non-zero entry in each section of $\boldsymbol{\beta}$ being uniformly distributed among the possible choices. Under the same distributional assumption, the final hard-decision step of the algorithm (3.11) is the

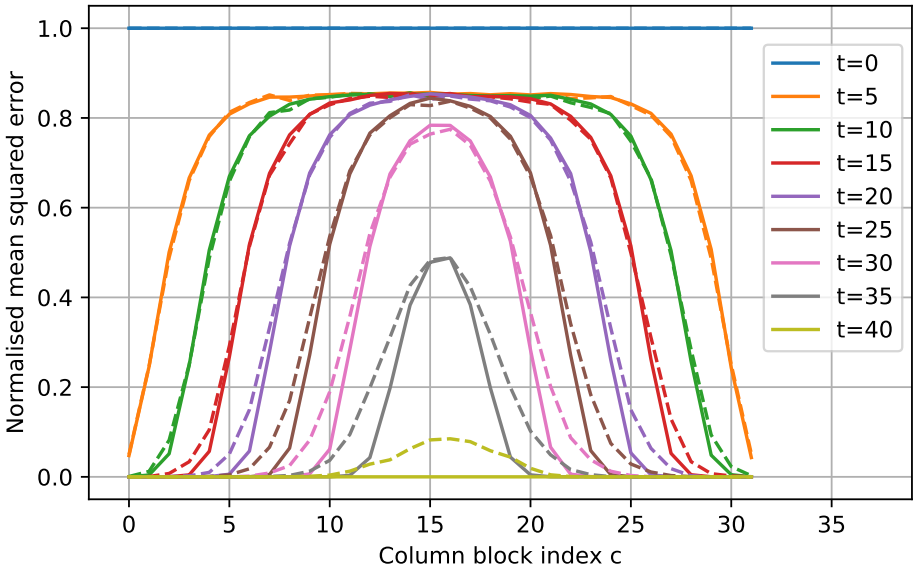


Figure 3.2: NMSE $\frac{\|\boldsymbol{\beta}_{\mathsf{c}}^t - \boldsymbol{\beta}_{\mathsf{c}}\|^2}{L/\mathsf{C}}$ vs. column block index $\mathsf{c} \in [\mathsf{C}]$ for several iteration numbers. The spatially coupled complex SPARC is defined via an $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix and has parameters $R = 3.1$ bits, $C = 4$ bits, $M = 256$, $L = 2048$ and $n = 5291$. The solid lines are the SE predictions from (3.15), and the dashed lines are the average NMSE over 100 instances of AMP decoding.

maximum a posteriori (MAP) estimator for $\boldsymbol{\beta}_{\mathrm{sec}(\ell)}$, i.e.,

$$\hat{\boldsymbol{\beta}}^T_{\mathrm{sec}(\ell)} = \underset{\boldsymbol{b} \in \mathcal{B}_{M,K}}{\arg\max}\ \mathbb{P}\Big(\boldsymbol{b} \,\Big|\, \boldsymbol{s}^{T-1}_{\mathrm{sec}(\ell)} = \boldsymbol{b} + \sqrt{\tilde{\boldsymbol{\tau}}^{T-1}_{\mathrm{sec}(\ell)}} \odot \boldsymbol{u}_{\mathrm{sec}(\ell)}\Big), \quad \text{for } \ell \in [L]. \tag{3.13}$$

State evolution Under the distributional property described above, the normalised mean square error (NMSE) between the true message vector and its estimate in each iteration of the AMP decoder can be predicted using a deterministic recursion called state evolution. For a modulated complex SPARC defined by base matrix $\boldsymbol{W} \in \mathbb{R}_+^{\mathsf{R}\times\mathsf{C}}$, and a complex AWGN channel with noise variance σ^2, state evolution (SE) iteratively defines vectors $\boldsymbol{\gamma}^t, \boldsymbol{\phi}^t \in \mathbb{R}^{\mathsf{R}}$ and $\boldsymbol{\tau}^t, \boldsymbol{\psi}^t \in \mathbb{R}^{\mathsf{C}}$ as follows. Initialise $\psi^0_{\mathsf{c}} = 1$ for $\mathsf{c} \in [\mathsf{C}]$, and for $t = 0, 1, \ldots$, compute

$$\gamma^t_{\mathsf{r}} = \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}} \psi^t_{\mathsf{c}}, \qquad \phi^t_{\mathsf{r}} = \sigma^2 + \gamma^t_{\mathsf{r}}, \qquad \mathsf{r} \in [\mathsf{R}], \tag{3.14}$$

$$\tau^t_{\mathsf{c}} = \frac{R/2}{\ln(KM)} \left[\frac{1}{\mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\phi^t_{\mathsf{r}}}\right]^{-1}, \qquad \psi^{t+1}_{\mathsf{c}} = 1 - \mathcal{E}(\tau^t_{\mathsf{c}}), \qquad \mathsf{c} \in [\mathsf{C}], \tag{3.15}$$

where $\mathcal{E}(\tau)$ is defined as follows for $\tau > 0$:

$$\mathcal{E}(\tau) = \mathbb{E}\left[\frac{\sum_{k=1}^K \Re[c_k] \cdot e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_k}\right]}}{\sum_{a=1}^K e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_a}\right]} + \sum_{j=2}^M \sum_{b=1}^K e^{\frac{1}{\sqrt{\tau}}\Re[U_j\,\overline{c_b}]}}\right], \tag{3.16}$$

with $U_1, \ldots, U_M \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, 2)$.

As illustrated in Fig. 3.2, the state evolution parameters $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$ closely track the NMSE of each block of the message vector, i.e., $\psi_{\mathsf{c}}^t \approx \frac{\|\boldsymbol{\beta}_{\mathsf{c}}^t - \boldsymbol{\beta}_{\mathsf{c}}\|^2}{L/\mathsf{C}}$ for $\mathsf{c} \in [\mathsf{C}]$. Similarly, the parameters $\{\phi_{\mathsf{r}}^t\}_{\mathsf{r}\in[\mathsf{R}]}$ closely track the block-wise variance of the modified residual vector $\boldsymbol{z}^t$, i.e. $\phi_{\mathsf{r}}^t \approx \frac{\|\boldsymbol{z}_{\mathsf{r}}^t\|^2}{n/\mathsf{R}}$ for $\mathsf{r} \in [\mathsf{R}]$.

For $t \geq 0$, the vectors $\widetilde{\boldsymbol{v}}^t \in \mathbb{R}^n$, $\widetilde{\boldsymbol{\tau}}^t \in \mathbb{R}^{LM}$, and the the matrix $\widetilde{\boldsymbol{Q}}^t \in \mathbb{R}^{n\times LM}$ in the AMP decoder (3.9) all have a block-wise structure, with their entries defined as follows. For $i \in [n]$ and $j \in [LM]$,

$$\widetilde{v}_i^t = \frac{\gamma_{\mathsf{r}(i)}^t}{\phi_{\mathsf{r}(i)}^{t-1}}, \qquad \widetilde{\tau}_j^t = \tau_{\mathsf{c}(j)}^t, \qquad \widetilde{Q}_{ij}^t = \frac{\tau_{\mathsf{c}(j)}^t}{\phi_{\mathsf{r}(i)}^t}, \tag{3.17}$$

where we recall that $\mathsf{r}(i)$ and $\mathsf{c}(j)$ denote the row and column block index of the i-th row entry and j-th column entry respectively. The vector $\widetilde{\boldsymbol{v}}^0$ is defined to be all-zeros.

Online estimation of SE parameters In the AMP decoder (3.9), instead of computing the SE parameters (3.14)–(3.17) offline they can be estimated online using intermediate outputs from the AMP. In particular, the SE parameters $\{\gamma_{\mathsf{r}}^t\}_{\mathsf{r}\in[\mathsf{R}]}, \{\phi_{\mathsf{r}}^t\}_{\mathsf{r}\in[\mathsf{R}]}$ and $\{\tau_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$, which are needed to compute $\widetilde{\boldsymbol{v}}^t$, $\widetilde{\boldsymbol{\tau}}^t$ and $\widetilde{\boldsymbol{Q}}^t$, can be estimated online as follows. For $\mathsf{r} \in [\mathsf{R}]$ and $\mathsf{c} \in [\mathsf{C}]$,

$$\hat{\gamma}_{\mathsf{r}}^t = \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}} \bigg(1 - \frac{\|\boldsymbol{\beta}_{\mathsf{c}}^t\|^2}{L/\mathsf{C}}\bigg), \tag{3.18}$$

$$\hat{\phi}_{\mathsf{r}}^t = \begin{cases} \sigma^2 + \hat{\gamma}_{\mathsf{r}}^t & \text{if the decoder knows } \sigma^2, \\ \frac{\|\boldsymbol{z}_{\mathsf{r}}^t\|^2}{n/\mathsf{R}} & \text{otherwise,} \end{cases} \tag{3.19}$$

$$\hat{\tau}_{\mathsf{c}}^t = \frac{R/2}{\ln(KM)} \left[\frac{1}{\mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\hat{\phi}_{\mathsf{r}}^t}\right]^{-1}. \tag{3.20}$$

These online estimates are used for the numerical simulations in Section 3.5. The justification for these estimates comes from [1, Lemma 7.6], which shows that in the case of unmodulated real-valued SPARCs, the estimates (3.18)–(3.20) concentrate on their respective state evolution parameters. The arguments of [1, Lemma 7.6] can be extended to show similar concentration of the online estimates for modulated complex SPARCs, but we do not pursue this in this thesis.

AMP and SE for real-valued versus complex-valued SPARCs The AMP decoder for unmodulated ($K = 1$) complex SPARCs in (3.9)–(3.10) simplifies to the AMP decoder for unmodulated real-valued SPARCs in (2.7)–(2.8) if a real-valued design matrix $\boldsymbol{A}$ is used. Moreover, the state evolution for unmodulated complex SPARCs in (3.14)–(3.16) simplifies to the state evolution for unmodulated real-valued SPARCs in (2.9)–(2.11), except that in the definition of τ_{c}^t in (3.15), the "$R/2$" term is replaced with "R" for a rate R unmodulated

real-valued SPARC.[4] This implies that the predicted NMSE of the AMP decoder for a rate R unmodulated complex SPARC is the same as that for a rate $R/2$ unmodulated real-valued SPARC. This matches our intuition since a transmission rate R over a complex AWGN channel with signal-to-ratio P/σ^2 corresponds to a rate of $R/2$ over each of two independent (real) AWGN channels with signal-to-noise ratio P/σ^2.

3.4 Error performance analysis

3.4.1 Error criteria

A natural error criterion for a SPARC decoder is the section error rate (SER) defined in (2.17), which is the fraction of sections decoded in error. (For the AMP decoder, the hard decision estimate $\hat{\boldsymbol{\beta}}^T$ is given in (3.11).) In a modulated SPARC, a section is decoded in error when *either* the location or the value of the non-zero entry in that section is estimated incorrectly. Recall that each section corresponds to $\log_2 K + \log_2 M$ nats of information.

We can also measure decoding performance via the bit error rate (BER). Recall that $\log_2 M$ bits determine the location of the non-zero entry in a section of the message vector $\boldsymbol{\beta}$, and $\log_2 K$ bits determine its value. If the location of the non-zero entry is estimated incorrectly, on average half of the $\log_2 M$ bits will be decoded wrongly due to the uniform mapping of bits to location. If the value is estimated incorrectly, on average less than half of the $\log_2 K$ bits will be decoded wrongly because Gray coding is used to map bits to constellation symbols.

A third performance measure is the frame error rate (FER), which is based on whether the entire SPARC codeword is decoded correctly. Since an error in decoding any of the sections leads to a codeword error, the FER is at least as large as the SER. In particular, we have FER $\geq$ SER $\geq$ BER.

In the following sections, we first show that the SER of the AMP decoder is upper bounded by a constant times its normalised mean squared error (NMSE). This result (Lemma 3.4.1) implies that an an upper bound on the SER can be obtained from an upper bound on the state evolution parameters $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$ (which predict the block-wise NMSE). Our main technical result (Proposition 3.4.1) provides such an upper bound on $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$. Using this bound, we show that in the large system limit, the state evolution recursion for complex SPARCs with K-ary PSK modulation (Corollary 3.4.1) is the *same* for any fixed value of K, including $K = 1$ (unmodulated). Therefore, the same arguments that prove that unmodulated SPARCs are capacity achieving with AMP decoding also imply that K-PSK modulated SPARCs are capacity

[4] One can also define τ_{c}^t in (3.15) using "R" instead "$R/2$" for complex SPARCs and change the definition of the functions η in (3.10) and $\mathcal{E}(\tau)$ in (3.16) accordingly. With this alternative definition of τ_{c}^t, the interpretation of the arguments to the function η in (3.12) would be slightly different: the first argument $\boldsymbol{s}$ would be approximately distributed as $\boldsymbol{\beta} + \sqrt{\tilde{\boldsymbol{\tau}}} \odot \boldsymbol{u}$, with $u_1, \ldots, u_{LM}$ i.i.d. $\sim \mathcal{CN}(0,1)$ (instead of i.i.d. $\sim \mathcal{CN}(0,2)$). The definition of τ_{c}^t in (3.15) using "$R/2$" makes the comparison of the AMP and SE equations between complex and real-valued SPARCs straightforward.

achieving with AMP decoding, for any fixed K (Theorems 2 and 3).

3.4.2 Bounding the section error rate

Lemma 3.4.1 (Bounding SER in terms of NMSE)**.** *Consider a complex SPARC with K-ary PSK modulation, with K being a power of 2. Let the AMP decoder be run for T iterations, with $\boldsymbol{\beta}^T$ being the final "soft-decision" estimate produced according to* (3.9)*, and $\hat{\boldsymbol{\beta}}^T$ the final hard-decision estimate produced according to* (3.11)*. Let the SER corresponding to $\hat{\boldsymbol{\beta}}^T$ be defined as in* (2.17)*. Then the SER can be bounded in terms of the normalised mean squared error $\frac{1}{L}\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2$ as follows:*

$$SER \leq \begin{cases} 4 \cdot \frac{\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2}{L} & \text{if } K = 1, 2, 4, \\ \sin^{-4}(\frac{\pi}{K}) \cdot \frac{\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2}{L} & \text{if } K \geq 8. \end{cases} \tag{3.21}$$

The proof is given in Section 3.6.1.

Since the NMSE $\frac{1}{L}\|\boldsymbol{\beta}^t - \boldsymbol{\beta}\|^2$ is predicted by the state evolution quantity $\frac{1}{\mathsf{C}}\sum_{\mathsf{c}} \psi_{\mathsf{c}}^t$, Lemma 3.4.1 implies that an upper bound on the parameters $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$ would give an upper bound on the SER after iteration t. The following proposition gives such an upper bound for $\{\psi_{\mathsf{c}}^t\}_{\mathsf{c}\in[\mathsf{C}]}$.

Proposition 3.4.1 (Bounding the state evolution predicted NMSE)**.** *Consider any base matrix $\boldsymbol{W}$ that satisfies $\xi_1 \leq \frac{1}{\mathsf{R}}\sum_{\mathsf{r}'} W_{\mathsf{r}'\mathsf{c}} \leq \xi_2$ and $\xi_1 \leq \frac{1}{\mathsf{C}}\sum_{\mathsf{c}'} W_{\mathsf{r}\mathsf{c}'} \leq \xi_2$ for some universal positive constants ξ_1, ξ_2, for all $\mathsf{r} \in [\mathsf{R}], \mathsf{c} \in [\mathsf{C}]$. Let*

$$\nu_{\mathsf{c}}^t \coloneqq \frac{1}{\tau_{\mathsf{c}}^t \cdot \ln(KM)} = \frac{2}{R \cdot \mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{r}\mathsf{c}}}{\phi_{\mathsf{r}}^t}, \quad \text{for } \mathsf{c} \in [\mathsf{C}]. \tag{3.22}$$

Then,

$$\frac{2}{R} \cdot \frac{\xi_1}{\sigma^2 + 2\xi_2} \leq \nu_{\mathsf{c}}^t \leq \frac{2}{R} \cdot \frac{\xi_2}{\sigma^2}, \tag{3.23}$$

and for K being a power of 2, sufficiently large M and any $\delta \in (0, \frac{1}{2})$,

$$\psi_{\mathsf{c}}^{t+1} \leq f_{K,M} \cdot \mathbb{1}\{\nu_{\mathsf{c}}^t > 2 + \delta\} + (1 + h_{K,M}) \cdot \mathbb{1}\{\nu_{\mathsf{c}}^t \leq 2 + \delta\}, \quad \text{for } \mathsf{c} \in [\mathsf{C}]. \tag{3.24}$$

The non-negative scalars $f_{K,M}$ and $h_{K,M}$ are bounded as:

$$f_{K,M} \leq \frac{(KM)^{-\alpha_{1K}\delta^2}}{\delta\sqrt{\ln(KM)}}, \qquad h_{K,M} \leq \frac{(KM)^{-\alpha_{2K}\nu_{\mathsf{c}}^t}}{\sqrt{\nu_{\mathsf{c}}^t \ln(KM)}}, \tag{3.25}$$

where α_{1K} and α_{2K} are positive constants depending only on K and the bounds of ν_{c}^t in (3.23)*. Exact expressions for $f_{K,M}, h_{K,M}$ are given in Remark 3.4.2 below.*

Remark 3.4.1. The variable ν_{c}^t can be regarded as a measure of the "signal-to-noise ratio" of column block c after iteration t. When ν_{c}^t exceeds $2+\delta$, the predicted NMSE of column block c

after iteration $t+1$, denoted by ψ_{c}^{t+1}, is upper bounded by $f_{K,M}$, which can be made arbitrarily small as $M \to \infty$ (Remark 3.4.3). On the other hand, if ν_{c}^t is less than $2+\delta$, ψ_{c}^{t+1} is upper bounded by $1+h_{K,M}$, which tends to 1 as $M \to \infty$.

Remark 3.4.2. The expressions for the scalars $f_{K,M}$ and $h_{K,M}$ when K is a power of 2, are as follows. For $K = 1, 2$ and 4,

$$f_{K,M},\ h_{K,M} = \begin{cases} \frac{M^{-\kappa_1 \delta^2}}{\delta\sqrt{\ln M}}, & 0, & \text{if } K=1 \\ \frac{(2M)^{-\kappa_2\delta^2}}{\delta\sqrt{\ln(2M)}}, & \frac{(2M)^{-\nu_{\mathsf{c}}^t/2}}{\sqrt{2\pi\nu_{\mathsf{c}}^t \ln(2M)}}, & \text{if } K=2 \\ \frac{(4M)^{-\kappa_3\delta^2}}{\delta\sqrt{\ln(4M)}}, & \frac{2(4M)^{-\nu_{\mathsf{c}}^t/2}}{\sqrt{2\pi\nu_{\mathsf{c}}^t \ln(4M)}}, & \text{if } K=4, \end{cases} \tag{3.26}$$

and for $K \geq 8$,

$$f_{K,M} = \frac{(1+\cot(\frac{2\pi}{K}))(KM)^{-\frac{\kappa_4\delta^2}{(1+\cot(\frac{2\pi}{K}))^2}}}{\delta\sqrt{\ln(KM)}} + K(KM)^{-2(2+\delta^2)\sin^2(\frac{\pi}{K})}, \tag{3.27}$$

$$h_{K,M} = \frac{2(1+\cot(\frac{2\pi}{K}))(KM)^{-\frac{\nu_{\mathsf{c}}^t}{2(1+\cot(\frac{2\pi}{K}))^2}}}{\sqrt{2\pi\nu_{\mathsf{c}}^t \ln(KM)}}, \tag{3.28}$$

where κ_1 to κ_4 are universal positive constants, not depending on K or M.

Remark 3.4.3. From (3.25)–(3.28), it is clear that for any fixed K and $\delta \in (0, \frac{1}{2})$, both $f_{K,M}$ and $h_{K,M}$ approach 0 as $M \to \infty$. On the other hand, for any fixed M and $\delta \in (0, \frac{1}{2})$, both $f_{K,M}$ and $h_{K,M}$ increase with K, for $K \geq 8$. Therefore, the upper bound on the predicted block-wise NMSE ψ_{c}^{t+1} in (3.24) and the upper bound on the SER in (3.21) both increase with K, when $K \geq 8$. This is consistent with the fact that when M is fixed, the amount of information transmitted per section increases with K.

If we consider increasing values of K, one can ask: how fast can K grow with M such that $f_{K,M}$ and $h_{K,M}$ approach 0 as K, M both approach infinity? Using the expressions in (3.25)–(3.28), we deduce that $f_{K,M}$ and $h_{K,M}$ approach 0 for any fixed $\delta \in (0, \frac{1}{2})$ if $\lim_{K,M\to\infty} |\frac{M}{g(K)}| = \infty$ (i.e., M dominates $g(K)$ asymptotically), where

$$g(K) = \frac{1}{K}\max\left(\left[1+\cot\left(\frac{2\pi}{K}\right)\right]^{\frac{[1+\cot(\frac{2\pi}{K})]^2}{\kappa_4\delta^2}},\ \left[1+\cot\left(\frac{2\pi}{K}\right)\right]^{\frac{2[1+\cot(\frac{2\pi}{K})]^2}{\min_{\mathsf{c}}\nu_{\mathsf{c}}^t}},\ K^{\frac{1}{2(2+\delta^2)\sin^2(\frac{\pi}{K})}}\right). \tag{3.29}$$

The proof of Proposition 3.4.1 is given in Section 3.6.2. A proof sketch is presented here to highlight the main ideas.

Proof Sketch. Recall from (3.15) that $\psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(\tau_{\mathsf{c}}^t)$. The upper bound on ψ_{c}^{t+1} in (3.24) is proved by obtaining two lower bounds for $\mathcal{E}(\tau_{\mathsf{c}}^t)$, one which holds for all values of $\nu_{\mathsf{c}}^t > 0$, and

the other which holds for $\nu_{\mathsf{c}}^t > 2$. In particular, we show that

$$\mathcal{E}(\tau_{\mathsf{c}}^t) \geq \begin{cases} -h_{K,M}, & \text{for } \nu_{\mathsf{c}}^t > 0, \\ 1 - f_{K,M}, & \text{for } \nu_{\mathsf{c}}^t > 2 + \delta. \end{cases} \tag{3.30}$$

For the case of $K = 1$ (no modulation), the result was proved in [57]. The proof for $K \geq 2$ is significantly more challenging as the $\mathcal{E}(\tau_{\mathsf{c}}^t)$ term defined in (3.16) can be negative. We discuss the $K \geq 2$ case below.

From (3.16) , we observe that $\mathcal{E}(\tau)$ is an expectation over M i.i.d. $\mathcal{CN}(0, 2)$ random variables $U_1, \ldots, U_M$. Using the tower property we write

$$\mathcal{E}(\tau) = \mathbb{E}_{U_1} \mathbb{E}_{U_2,\ldots,U_M} \left[\frac{\sum_{k=1}^K \Re[c_k] \cdot e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_k}\right]}}{\sum_{a=1}^K e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_a}\right]} + \sum_{j=2}^M \sum_{b=1}^K e^{\frac{1}{\sqrt{\tau}}\Re[U_j\,\overline{c_b}]}} \,\middle|\, U_1 \right], \tag{3.31}$$

noting that $U_2, \ldots, U_M$ only appear in the denominator. The outer expectation over U_1 is split into four terms, each of which integrate over different ranges of U_1^R and U_1^I, the independent real and imaginary parts of U_1.

$$\begin{aligned} \mathcal{E}(\tau) &= \int_{\underline{u}}^{\infty} \int_{\underline{u}}^{\infty} p(u^R) p(u^I) \mathbb{E}_{U_2,\ldots,U_M}[\ldots] du^R du^I + \int_{-\infty}^{\underline{u}} \int_{-\infty}^{\underline{u}} p(u^R) p(u^I) \mathbb{E}_{U_2,\ldots,U_M}[\ldots] du^R du^I \\ &\quad + \int_{\underline{u}}^{\infty} \int_{-\infty}^{\underline{u}} p(u^R) p(u^I) \mathbb{E}_{U_2,\ldots,U_M}[\ldots] du^R du^I + \int_{-\infty}^{\underline{u}} \int_{\underline{u}}^{\infty} p(u^R) p(u^I) \mathbb{E}_{U_2,\ldots,U_M}[\ldots] du^R du^I \\ &= I_1 + I_2 + I_3 + I_4, \end{aligned} \tag{3.32}$$

where $\underline{u}$ is an arbitrary parameter which splits the range of the integration.

We argue that for a suitably chosen value of $\underline{u}$ (which depends on K, M but not on ν_{c}^t), the $\mathbb{E}_{U_2,\ldots,U_M}[\ldots]$ term is non-negative for $U_1^R \geq \underline{u}$ and $U_1^I \geq \underline{u}$, and hence $I_1 \geq 0$. For the terms I_2 to I_4, we first show that $\mathbb{E}_{U_2,\ldots,U_M}[\ldots] \geq -1$, and then use tail bounds for standard normals to obtain $I_2 + I_3 + I_4 \geq -h_{K,M}$. Thus we obtain the lower bound $\mathcal{E}(\tau_{\mathsf{c}}^t) \geq -h_{K,M}$ for all $\nu_{\mathsf{c}}^t > 0$.

For $\nu_{\mathsf{c}}^t > 2$, we again split the integral into the ranges in (3.32), but choose $\underline{u}$ to depend on ν_{c}^t as well as K, M. We again use $\mathbb{E}_{U_2,\ldots,U_M}[\ldots] \geq -1$ and tail bounds for standard normals to obtain a lower bound of the form $I_2 + I_3 + I_4 \geq -B_1 \cdot f_{K,M}$ for sufficiently large M and positive constant B_1. We then obtain a lower bound for I_1 which takes the form $I_1 \geq 1 - B_2 \cdot f_{K,M}$ for sufficiently large M and positive constant B_2. Combining the results, we obtain the lower bound on $\mathcal{E}(\tau_{\mathsf{c}}^t)$ for the $\nu_{\mathsf{c}}^t > 2$ case. We therefore have both the lower bounds in (3.30). □

In addition to the upper bound on the state evolution parameter ψ_{c}^{t+1} given in Proposition 3.4.1, a corresponding lower bound can be derived for the $\nu_{\mathsf{c}}^t < 2$ case.

Proposition 3.4.2. *Under the same conditions as Proposition 3.4.1, for sufficiently large M,*

and any $\tilde{\delta} \in (0,1)$,

$$\psi_{\mathsf{c}}^{t+1} \geq \left(1 - M^{-\alpha_{3K}\tilde{\delta}^2}\right) \cdot \mathbb{1}\{\nu_{\mathsf{c}}^t < 2 - \tilde{\delta}\}, \tag{3.33}$$

where α_{3K} *is a positive constant depending only on* K *and the bounds of* ν_{c}^t *in* (3.23).

The above result was proved for the unmodulated ($K = 1$) case in [1, Lem. 4.1] (see also Lemma 2.3.1). To prove this lower bound for the general $K \geq 1$ case, we use similar high-level arguments as in the proof of [1, Lem. 4.1], as well as specific techniques used in the proof of the upper bound in Proposition 3.4.1 for the $K \geq 1$ case. The proof is given in Section 3.6.4.

Asymptotic state evolution

Proposition 3.4.1 together with Proposition 3.4.2 immediately yields the following asymptotic characterisation of the state evolution recursion for $M \to \infty$.

Corollary 3.4.1 (Asymptotic state evolution)**.** *For any base matrix* $\boldsymbol{W}$ *satisfying the conditions in Proposition 3.4.1, the state evolution recursion in* (3.14)–(3.16) *simplifies to the following as* $M \to \infty$*, for any fixed* $K \geq 1$*. Initialise* $\bar{\psi}_{\mathsf{c}}^0 = 1$ *for* $\mathsf{c} \in [\mathsf{C}]$*, and for* $t = 0, 1, \ldots$*, compute*

$$\bar{\phi}_{\mathsf{r}}^t \;=\; \sigma^2 + \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}} \bar{\psi}_{\mathsf{c}}^t, \qquad \mathsf{r} \in [\mathsf{R}], \tag{3.34}$$

$$\bar{\psi}_{\mathsf{c}}^{t+1} \;=\; \mathbb{1}\left\{ \frac{1}{\mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\bar{\phi}_{\mathsf{r}}^t} \leq R \right\}, \qquad \mathsf{c} \in [\mathsf{C}], \tag{3.35}$$

where $\bar{\phi}, \bar{\psi}$ *indicate asymptotic values.*

Remark 3.4.4. Though the state evolution parameters in (3.14)–(3.16) depend on the modulation factor K, the asymptotic values of these parameters in (3.34)–(3.35) do not. Therefore, as $M \to \infty$, the predicted per-iteration NMSE of the AMP decoder for complex SPARCs with K-ary PSK modulation is the same for any finite K, including $K = 1$ (unmodulated).

Remark 3.4.5 (Complex versus real asymptotic SE)**.** The only difference between the asymptotic state evolution in (3.34)–(3.35) for modulated complex SPARCs and that for unmodulated real-valued SPARCs in (2.22)–(2.23) is that R within the indicator in (3.35) is replaced by $2R$ in (2.23). This matches our intuition since a rate of R nats over a complex AWGN channel corresponds to a rate of $R/2$ nats/dimension (see also last paragraph of Section 3.3.1).

We note that the lower bound in Proposition 3.4.2 is not required for Theorems 2 and 3, which are the main results of this chapter (see Section 3.4.3); only the upper bounds in Proposition 3.4.1 are required. That is, Theorems 2 and 3 does not require the equality in (3.35), but only an upper bound on $\bar{\psi}^{t+1}$, which tracks the NMSE of the AMP decoder.

3.4.3 PSK-modulated SPARCs are capacity achieving

The asymptotic state evolution equations in (3.34)–(3.35) have been analysed for two choices of base matrix $\boldsymbol{W}$: for a suitable (ω, Λ, ρ) base matrix in Chapter 2 (which has appeared in [1,2]), and for the base matrix corresponding to an exponentially decaying power allocation in [22]. These results show that in both cases, for any fixed $R < C$ we have $\bar{\psi}_{\mathsf{c}}^T = 0$ for $\mathsf{c} \in [\mathsf{C}]$, where the number of iterations T is a finite value depending on the rate gap from capacity. That is, the state evolution equations predict reliable decoding in the large system limit for $R < C$.

We know that the asymptotic equations (3.34)–(3.35) hold for K-PSK modulated complex SPARCs, for any $K \geq 1$ (Remarks 3.4.4 and 3.4.5). We can therefore directly use the asymptotic state evolution results in Chapter 2 and [22] to argue that K-PSK modulated SPARCs are asymptotically capacity achieving for any finite K, with the base matrices used in Chapter 2 and [22]. Theorem 2 gives this result for modulation applied to spatially coupled SPARCs defined via an (ω, Λ, ρ) base matrix. Theorem 3 gives a similar result for SPARCs with exponentially decaying power allocation. We require some definitions to state the theorems.

For an (ω, Λ, ρ) base matrix, let $\vartheta = 1 + \frac{\omega - 1}{\Lambda}$, and

$$R^\star = \frac{1}{\vartheta} \ln(1 + \vartheta\, \mathsf{snr}), \tag{3.36}$$

where $\mathsf{snr} = \frac{P}{\sigma^2}$. We will consider rates $R < R^\star$ (in nats), noting that $C > R^\star > \frac{C}{\vartheta}$, where

$$C = \ln(1 + \mathsf{snr}). \tag{3.37}$$

Observe that $\vartheta \to 1$ as $\frac{\omega}{\Lambda} \to 0$, and hence $R^\star$ can be made arbitrarily close to C for any fixed ω by choosing Λ to be sufficiently large. Finally, let

$$\omega^\star = \frac{\vartheta\, \mathsf{snr}^2}{(1 + \vartheta\, \mathsf{snr})(R^\star - R)}. \tag{3.38}$$

Theorem 2 (K-PSK modulated SPARCs with spatial coupling are capacity achieving)**.** *For any $R < C$, let $\boldsymbol{W}$ be an (ω, Λ, ρ) base matrix with parameters chosen such that $R^\star > R$, $\omega > \omega^\star$ and $\rho = \min\{\frac{1}{2}, \frac{R^\star - R}{3\mathsf{snr}}\}$, where $R^\star, \omega^\star$ are defined in* (3.36) *and* (3.38)*. Fix K to be a power of 2. Let $\{\mathcal{S}_n\}$ be a sequence of rate R, K-PSK modulated SPARCs (indexed by code length n), with $\mathcal{S}_n$ defined via an $n \times LM$ design matrix constructed from the base matrix $\boldsymbol{W}$. Let $SER(\mathcal{S}_n) := \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{\beta}}_{sec(\ell)}^T \neq \boldsymbol{\beta}_{sec(\ell)}\}$ denote the section error rate of the AMP decoder after T iterations where $T = \lceil \frac{\Lambda \omega^\star}{2\omega} \rceil$. Then*

$$\lim_{n \to \infty} SER(\mathcal{S}_n) = 0 \text{ almost surely,}$$

where the limit is taken with (L, M, n) all tending to infinity such that $R = L \ln(KM)/n$.

Proof. Recall the definition of asymptotic state evolution parameter $\bar{\psi}_{\mathsf{c}}^T$ from Corollary 3.4.1. Under the stated conditions on the parameters of the (ω, Λ, ρ) base matrix, Proposition 2.3.1 shows that

$$\bar{\psi}_{\mathsf{c}}^T = 0, \quad \text{for } \mathsf{c} \in [\mathsf{C}]. \tag{3.39}$$

Furthermore, [1, Thm. 2] implies that

$$\lim \frac{\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2}{L} = \frac{1}{\mathsf{C}} \sum_{\mathsf{c} \in [\mathsf{C}]} \bar{\psi}_{\mathsf{c}}^T \quad \text{almost surely,} \tag{3.40}$$

where the limit is taken with (L, M, n) all tending to infinity such that $R = L \ln(KM)/n$. The concentration result in (3.40) is shown for unmodulated real-valued SPARCs in [1], and the proof for K-PSK modulated complex SPARCs case is essentially the same. Combining (3.39) and (3.40) with Lemma 3.4.1 yields the statement of the theorem. □

For any rate $R < C$, we can choose the base matrix parameters ω and Λ using the method described in Remark 2.3.2 to satisfy the conditions of the theorem.

Theorem 3 (K-PSK modulated SPARCs with exponentially decaying power allocation are capacity achieving)**.** *For any $R < C$, let $\boldsymbol{W}$ be a $1 \times L$ base matrix corresponding to the exponentially decaying power allocation, i.e.,*

$$W_{1\ell} = LP \cdot \frac{e^{C/L} - 1}{1 - e^{-C}} \cdot e^{-C\ell/L}, \quad \ell \in [L]. \tag{3.41}$$

Fix K to be a power of 2. Let $\{\mathcal{S}_n\}$ be a sequence of rate R, K-PSK modulated SPARCs (indexed by code length n), with $\mathcal{S}_n$ defined via an $n \times LM$ design matrix constructed from the base matrix $\boldsymbol{W}$. Let $SER(\mathcal{S}_n) := \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{\beta}}_{sec(\ell)}^T \neq \boldsymbol{\beta}_{sec(\ell)}\}$ denote the section error rate of the AMP decoder after T iterations where $T = \lceil \frac{C}{\ln(C/R)} \rceil$. Then

$$\lim_{n \to \infty} SER(\mathcal{S}_n) = 0 \text{ almost surely,}$$

where the limit is taken with (L, M, n) all tending to infinity such that $R = L \ln(KM)/n$.

Proof. With the exponentially decaying allocation in (3.41), using $\mathsf{R} = 1$, $\mathsf{C} = L$, the asymptotic state evolution recursion in Corollary 3.4.1 reduces to

$$\bar{\psi}_{\ell}^{t+1} = \mathbb{1}\left\{ W_{1\ell} \leq R\Big(\sigma^2 + \frac{1}{L} \sum_{j=1}^{L} W_{1j}\, \bar{\psi}_j^t\Big) \right\}, \quad \ell \in [L]. \tag{3.42}$$

This recursion is analysed in [22, Lemmas 1 and 2], where it is shown that

$$\lim \frac{1}{L} \sum_{\ell=1}^{L} W_{1\ell}\, \bar{\psi}_\ell^T = 0, \tag{3.43}$$

for $T = \lceil \frac{C}{\ln(C/R)} \rceil$. Furthermore, [1, Thm. 2] implies that

$$\lim \left| \frac{\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2}{L} - \frac{1}{L} \sum_{\ell \in [L]} \bar{\psi}_\ell^T \right| = 0 \quad \text{almost surely.} \tag{3.44}$$

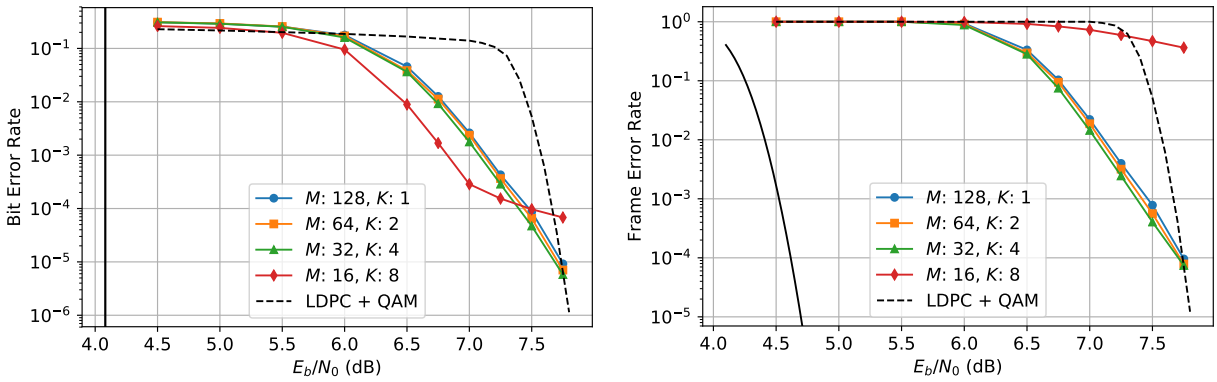
In both (3.43) and (3.44), the limit is taken with (L, M, n) all tending to infinity such that $R = L \ln(KM)/n$. The concentration result in (3.44) is shown for unmodulated real-valued SPARCs in [1], and the proof for K-PSK modulated complex SPARCs is essentially the same. As $L \to \infty$, the base matrix entries $\{W_{1\ell}\}_{\ell \in [L]}$ in (3.41) are bounded above and below by strictly positive universal constants. Therefore, (3.43) implies that $\frac{1}{L} \sum_{\ell \in [L]} \bar{\psi}_\ell^T = 0$. Using this in (3.44) and then invoking Lemma 3.4.1 yields the statement of the theorem. □

Theorems 2 and 3 correspond to two different choices of base matrices for which $\bar{\psi}_{\mathsf{c}}^T = 0$ for $\mathsf{c} \in [\mathsf{C}]$. The key requirement for a PSK-modulated complex SPARC design to be capacity achieving is that the underlying base matrix satisfies $\frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} \bar{\psi}_{\mathsf{c}}^T = 0$, for all fixed $R < C$. Here T is the final iteration, which in the examples above is determined by the gap from capacity $(C - R)$. Whenever we have a base matrix whose asymptotic state evolution recursion (3.34)–(3.35) satisfies the above property, [1, Thm. 2] and Lemma 3.4.1 together imply that the design is capacity achieving in the large system limit.

3.5 Empirical error performance

In Figs. 3.3 and 3.4 we show the finite length error performance of PSK-modulated complex SPARCs with AMP decoding via numerical simulations. The error performance is evaluated using bit error rate (BER) and frame error rate (FER) as they are common performance metrics of interest. For reference, we also simulate and plot the error performance of coded modulation schemes (LDPC + QAM) using the AFF3CT toolbox [154]. The LDPC codes used are from the DVB-S2 standard.

Within each figure, the SPARCs have the same rate R, code length n, and number of sections L. We vary the code parameters K and M while keeping their product KM constant, recalling that $R = \frac{L \log_2(KM)}{n}$ bits. As K increases from 1 (unmodulated) to 4, both the BER and the FER improve. At $K = 8$, the BER continues to improve at low values of E_b/N_0, but an error floor starts to appear at high E_b/N_0. However, the FER at $K = 8$ is significantly worse than that of $K = 1, 2, 4$ for all values of E_b/N_0. We expect that a much larger M is required to



(a) Bit error rate

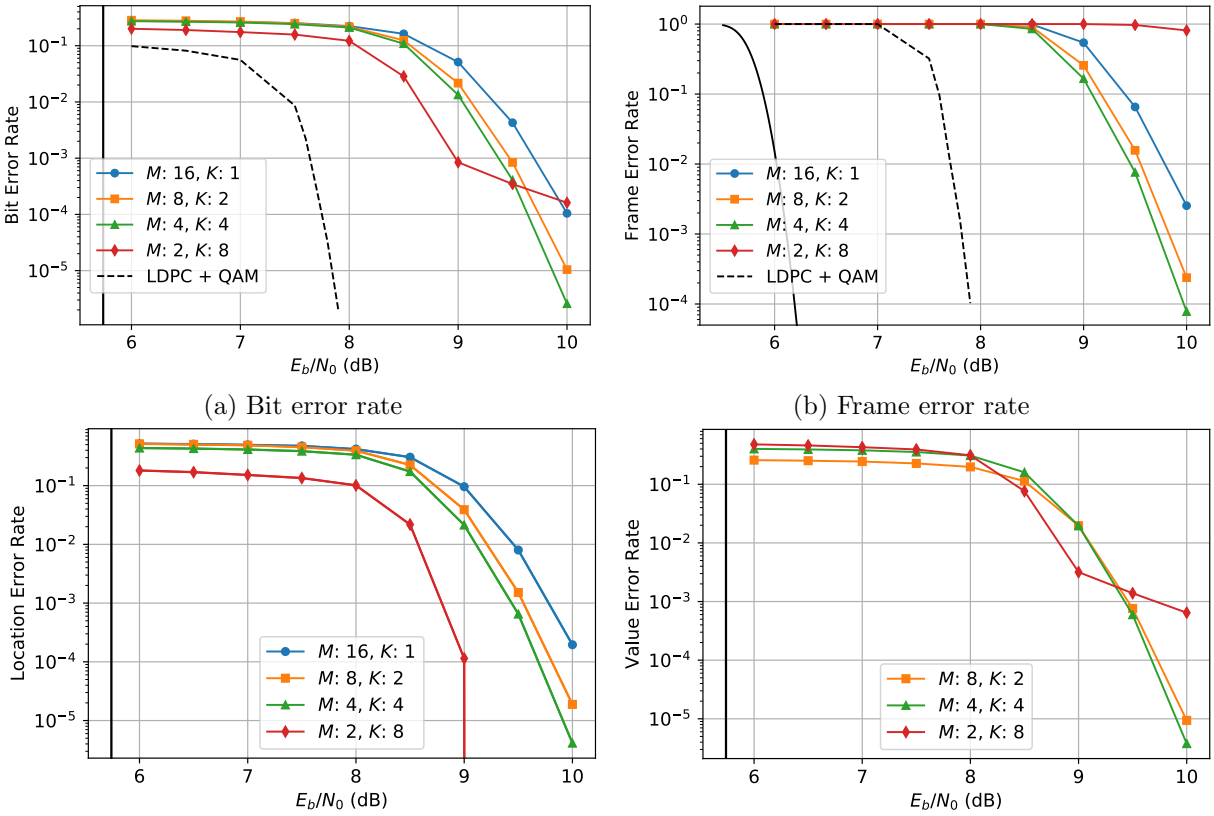
(b) Frame error rate

Figure 3.3: Error performance of K-ary PSK modulated complex SPARCs defined via a $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix. Code parameters: $R = 1.593$ bits/dimension, $L = 960$, code length $n = 2109$. Each curve represents a K and M pair with fixed $KM = 128$. The dashed lines show the performance of coded modulation: (6480, 16200) DVB-S2 LDPC + 256 QAM, frame length = 2025, overall rate = 1.6 bits/dimension. The solid black line in subplot (a) is the AWGN Shannon limit for $R = 1.6$ bits/dimension, and in subplot (b) it is the normal approximation to AWGN finite length error bound in [34].

achieve low FER with $K = 8$, but more investigation is required.

Recall that in a modulated SPARC, a section error occurs when either the location or value (or both) of the non-zero entry in the section is decoded incorrectly. Figs. 3.4c and 3.4d show the location error rate (fraction of sections where the location of the non-zero entry was decoded in error) and the value error rate (fraction of sections where the value of the non-zero entry was decoded in error) from the same set of simulations used to plot Figs. 3.4a and 3.4b. We notice that with KM held fixed, the location error rate consistently improves as K increases and M decreases. We also notice that when $E_b/N_0 > 9$ dB, all errors in decoding the $M = 2, K = 8$ modulated complex SPARC were due to value errors. Since bit errors arise from a combination of both location and value errors (see beginning of Section 3.4), the location error rate and value error rate curves help us understand the shape of the BER curve in Fig. 3.4a.

Implementation details For the simulations, a discrete Fourier transform (DFT) based design matrix was used instead of a Gaussian one. This enables the matrix-vector multiplications in the AMP decoder (3.9) to be computed via the fast Fourier transform (FFT), which significantly lowers the decoding complexity and memory requirement. The error performance of DFT based design matrices were found to be similar to that of Gaussian matrices for large matrix sizes. Our approach is similar to that discussed in Section 2.5.1 (and [22,56]) where Hadamard-based design matrices were used for unmodulated real-valued SPARCs. For the simulations, we also use the online estimates of the state evolution parameters described in (3.18)–(3.20). The code used for the simulations is available at [155].



(a) Bit error rate

(b) Frame error rate

(c) Location error rate

(d) Value error rate

Figure 3.4: Error performance of modulated complex SPARCs defined via a $(\omega = 6, \Lambda = 32, \rho = 0)$ base matrix. Code parameters: $R = 1.99$ bits/dimension, $L = 2688$, code length $n = 2701$. Each curve represents a K and M pair with fixed $KM = 16$. The dashed lines show the performance of coded modulation: (10800, 16200) DVB-S2 LDPC + 64 QAM, frame length = 2700, overall rate = 2.0 bits/dimension. The solid black lines in subplots (a), (c) and (d) are the AWGN Shannon limit for $R = 2.0$ bits/dimension, and in subplot (b) it is the normal approximation to AWGN finite length error bound in [34].

3.5.1 Decoding complexity

The complexity of the AMP decoder is dominated by the two matrix-vector multiplications (which are replaced by FFTs) in (3.9), and the η function in (3.10). The complexity of the FFTs are $O(LM\log(LM))$ and the complexity of the η function is $O(LMK)$. Therefore, the overall complexity per iteration is $O(LM(\log(LM)+K))$. In each set of simulations, as K increases and M decreases, with the product KM kept constant, the overall decoding complexity decreases.

To compare the decoding complexities of unmodulated and modulated SPARCs with KM held fixed, denote the values of M for the two cases by M_{unmod} and M_{mod}, so that $M_{\text{unmod}} = KM_{\text{mod}}$. Then the ratio of decoding complexities is

$$\frac{\text{complexity for unmodulated SPARC}}{\text{complexity for modulated SPARC}} = K \cdot \frac{\log(LM_{\text{unmod}}) + 1}{\log(LM_{\text{unmod}}) + K - \log K}. \tag{3.45}$$

If $K \ll \log(LM)$, then modulation can reduce decoding complexity by nearly K times. For

example, in Fig. 3.3 the decoding complexity is reduced by approximately 3.8 times from ($K = 1, M = 128$) to ($K = 4, M = 32$) while the error performance improves.

The simulations above indicate that for a fixed code length n, rate R, and number of sections L, modulation can significantly reduce decoding complexity without sacrificing error performance. Modulation also allows more flexibility in the code design of SPARCs. For example, due to the rate equation $R = \frac{L \ln(KM)}{n}$, for a fixed code length n, number of sections L, and section size M, one can increase the rate of the SPARC by increasing the modulation parameter K, and this increase of K will not affect the decoding complexity if $K \ll \log(LM)$.

3.6 Proofs

3.6.1 Proof of Lemma 3.4.1

Recall that $\text{sec}(\ell) \coloneqq \{(\ell-1)M + 1, \ldots, \ell M\}$ for $\ell \in [L]$ and $\boldsymbol{\beta}_{\text{sec}(\ell)} \in \mathbb{C}^M$ denotes the ℓ-th section of the message vector $\boldsymbol{\beta} \in \mathbb{C}^{LM}$. We will show that if $\boldsymbol{\beta}_{\text{sec}(\ell)}$ is decoded in error after T iterations of AMP decoding, then the squared error of that section is lower bounded by a positive number that is a function of the modulation parameter K. More precisely, we show that

$$\hat{\boldsymbol{\beta}}^T_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)} \;\Rightarrow\; \|\boldsymbol{\beta}^T_{\text{sec}(\ell)} - \boldsymbol{\beta}_{\text{sec}(\ell)}\|^2 \geq h(K), \tag{3.46}$$

where

$$h(K) = \begin{cases} \frac{1}{4} & \text{if } K = 1, 2, 4, \\ \sin^4(\frac{\pi}{K}) & \text{if } K \geq 8. \end{cases} \tag{3.47}$$

Then (3.46) implies

$$\text{SER} = \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\left\{\hat{\boldsymbol{\beta}}^T_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)}\right\} \leq \frac{1}{h(K)}\frac{1}{L}\sum_{\ell=1}^{L} \|\boldsymbol{\beta}^T_{\text{sec}(\ell)} - \boldsymbol{\beta}_{\text{sec}(\ell)}\|^2 = \frac{1}{h(K)} \cdot \frac{\|\boldsymbol{\beta}^T - \boldsymbol{\beta}\|^2}{L},$$

which is the required result after substituting in (3.47). We now prove the statements given in (3.46)–(3.47).

We denote the location index of the non-zero entry of $\boldsymbol{\beta}_{\text{sec}(\ell)}$ as $\text{sent}(\ell)$ (let's assume section ℓ is in column block $\mathsf{c} \in [\mathsf{C}]$). By symmetry of the PSK constellation, we can assume without loss of generality that the value of the non-zero entry is $c_K = +1$. Thats is, for $j \in \text{sec}(\ell)$,

$$\beta_j = \begin{cases} 1 & \text{if } j = \text{sent}(\ell), \\ 0 & \text{otherwise.} \end{cases} \tag{3.48}$$

Thus,

$$\|\boldsymbol{\beta}^T_{\text{sec}(\ell)} - \boldsymbol{\beta}_{\text{sec}(\ell)}\|^2 \geq |\beta^T_{\text{sent}(\ell)} - \beta_{\text{sent}(\ell)}|^2 = |\beta^T_{\text{sent}(\ell)} - 1|^2$$

$$
\begin{aligned}
&\overset{\text{(i)}}{=} \left| \sum_{k=1}^{K} c_k \cdot \frac{e^{\Re(\overline{s^{T-1}_{\text{sent}(\ell)}} c_k)/\tau_{\mathsf{c}}^{T-1}}}{\sum_{j' \in \text{sec}(\ell)} \sum_{k'=1}^{K} e^{\Re(\overline{s^{T-1}_{j'}} c_{k'})/\tau_{\mathsf{c}}^{T-1}}} - 1 \right|^2 \\
&\overset{\text{(ii)}}{=} \left[\sum_{k=1}^{K} \cos\left(\frac{2\pi k}{K}\right) \cdot p_k - 1 \right]^2 + \left[\sum_{k=1}^{K} \sin\left(\frac{2\pi k}{K}\right) \cdot p_k \right]^2 \\
&\geq \left[\sum_{k=1}^{K} \cos\left(\frac{2\pi k}{K}\right) \cdot p_k - 1 \right]^2, \qquad (3.49)
\end{aligned}
$$

where (i) is obtained using the expression for $\beta^T_{\text{sent}(\ell)}$ derived from (3.9), (3.10) and (3.17), and (ii) from substituting in $c_k = e^{\mathrm{j}2\pi k/K}$ and defining

$$
p_k \coloneqq \frac{e^{\Re(\overline{s^{T-1}_{\text{sent}(\ell)}} c_k)/\tau_{\mathsf{c}}^{T-1}}}{\sum_{j' \in \text{sec}(\ell)} \sum_{k'=1}^{K} e^{\Re(\overline{s^{T-1}_{j'}} c_{k'})/\tau_{\mathsf{c}}^{T-1}}}, \qquad k \in [K]. \qquad (3.50)
$$

Towards proving (3.46), assume that $\hat{\boldsymbol{\beta}}^T_{\text{sec}(\ell)} \neq \boldsymbol{\beta}_{\text{sec}(\ell)}$. Then from (3.11) we know that $\Re(\overline{s^{T-1}_{\text{sent}(\ell)}} c_K) \leq \Re(\overline{s^{T-1}_{j^*}} c_{k^*})$ for some $(j^*, k^*) \neq (\text{sent}(\ell), K)$. This gives us the following inequality.

$$
\begin{aligned}
p_K = \frac{e^{\Re(\overline{s^{T-1}_{\text{sent}(\ell)}} c_K)/\tau_{\mathsf{c}}^{T-1}}}{\sum_{j' \in \text{sec}(\ell)} \sum_{k'=1}^{K} e^{\Re(\overline{s^{T-1}_{j'}} c_{k'})/\tau_{\mathsf{c}}^{T-1}}} &\leq \frac{e^{\Re(\overline{s^{T-1}_{j^*}} c_{k^*})/\tau_{\mathsf{c}}^{T-1}}}{\sum_{j' \in \text{sec}(\ell)} \sum_{k'=1}^{K} e^{\Re(\overline{s^{T-1}_{j'}} c_{k'})/\tau_{\mathsf{c}}^{T-1}}} \\
&\leq 1 - \frac{e^{\Re(\overline{s^{T-1}_{\text{sent}(\ell)}} c_K)/\tau_{\mathsf{c}}^{T-1}}}{\sum_{j' \in \text{sec}(\ell)} \sum_{k'=1}^{K} e^{\Re(\overline{s^{T-1}_{j'}} c_{k'})/\tau_{\mathsf{c}}^{T-1}}} = 1 - p_K, \qquad (3.51)
\end{aligned}
$$

where the second inequality is obtained by noting that

$$
\sum_{j \in \text{sec}(\ell)} \sum_{k=1}^{K} \frac{e^{\Re(\overline{s^{T-1}_{j}} c_k)/\tau_{\mathsf{c}}^{T-1}}}{\sum_{j' \in \text{sec}(\ell)} \sum_{k'=1}^{K} e^{\Re(\overline{s^{T-1}_{j'}} c_{k'})/\tau_{\mathsf{c}}^{T-1}}} = 1.
$$

From (3.51) and the fact that $p_K \geq 0$, we deduce $0 \leq p_K \leq \frac{1}{2}$.

Continuing from (3.49), we obtain the required lower bounds on the squared error of section ℓ when it is decoded in error.

$$
\begin{aligned}
\left[\sum_{k=1}^{K} \cos\left(\frac{2\pi k}{K}\right) \cdot p_k - 1 \right]^2 &= \begin{cases} (1 \cdot p_1 - 1)^2 & \text{if } K = 1, \\ (-1 \cdot p_1 + 1 \cdot p_2 - 1)^2 & \text{if } K = 2, \\ (-1 \cdot p_2 + 1 \cdot p_4 - 1)^2 & \text{if } K = 4, \end{cases} \\
&\geq \frac{1}{4}, \qquad (3.52)
\end{aligned}
$$

where the last inequality is obtained using $0 \leq p_K \leq \frac{1}{2}$ and $p_k \geq 0$ for $k \in [K]$.

For the $K \geq 8$ case, first notice that $|\sum_k \cos(2\pi k/K) \cdot p_k| \leq 1$ since $p_k \geq 0$ for $k \in [K]$ and $\sum_k p_k \leq 1$. Moreover,

$$\begin{aligned}\sum_{k=1}^{K} \cos\Big(\frac{2\pi k}{K}\Big) \cdot p_k \;&\leq\; \cos\Big(\frac{2\pi K}{K}\Big) \cdot p_K + \max_{k\in\{1,\ldots,K-1\}} \Big\{ \cos\Big(\frac{2\pi k}{K}\Big)\Big\} \cdot \Big(\sum_{k=1}^{K-1} p_k\Big) \\ &\leq p_K + \cos\Big(\frac{2\pi}{K}\Big) \cdot (1 - p_K). \end{aligned} \tag{3.53}$$

Using this together with $|\sum_k \cos(2\pi k/K) \cdot p_k| \leq 1$, we obtain

$$\begin{aligned}\Big[\sum_{k=1}^{K} \cos\Big(\frac{2\pi k}{K}\Big) \cdot p_k - 1\Big]^2 &\geq \Big[(1-p_K)\Big(\cos(2\pi/K) - 1\Big)\Big]^2 \\ &\geq \frac{1}{4}\Big(1 - \cos(2\pi/K)\Big)^2 = \sin^4(\pi/K)\,. \end{aligned} \tag{3.54}$$

Here the second inequality is obtained using $0 \leq p_K \leq \frac{1}{2}$. Using (3.52) and (3.54) in (3.49) completes the proof of (3.46), and hence the lemma.

3.6.2 Proof of Proposition 3.4.1

We will prove the proposition for K — the size of the PSK constellation — being a power of 2.

We first obtain the lower and upper bounds of ν_{c}^t given in (3.23). To do this, we first show that the absolute value of $\mathcal{E}(\tau)$ defined in (3.16) is bounded. Indeed,

$$\begin{aligned}|\mathcal{E}(\tau)| &\leq \mathbb{E}\left[\left|\frac{\sum_{k=1}^{K} \Re[c_k] \cdot e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_k}\right]}}{\sum_{a=1}^{K} e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_a}\right]} + \sum_{j=2}^{M}\sum_{b=1}^{K} e^{\frac{1}{\sqrt{\tau}}\Re[U_j\overline{c_b}]}}\right|\right] \\ &\overset{\text{(i)}}{\leq} \mathbb{E}\left[\frac{\sum_{k=1}^{K} e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_k}\right]}}{\sum_{a=1}^{K} e^{\frac{1}{\tau}\Re\left[(1+\sqrt{\tau}U_1)\,\overline{c_a}\right]} + \sum_{j=2}^{M}\sum_{b=1}^{K} e^{\frac{1}{\sqrt{\tau}}\Re[U_j\overline{c_b}]}}\right] \\ &\leq 1, \end{aligned} \tag{3.55}$$

where (i) is obtained by $|\Re[c_k]\,| \leq 1$. Using the above result in (3.15) we deduce that

$$0 \leq \psi_{\mathsf{c}}^t \leq 2. \tag{3.56}$$

Now we obtain upper and lower bounds for ν_{c}^t,

$$\nu_{\mathsf{c}}^t \overset{\text{(i)}}{=} \frac{2}{R\,\mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\sigma^2 + \frac{1}{\mathsf{C}}\sum_{\mathsf{c}'} W_{\mathsf{rc}'}\psi_{\mathsf{c}'}^t} \overset{\text{(ii)}}{\leq} \frac{2}{R\,\mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\sigma^2} \overset{\text{(iii)}}{\leq} \frac{2}{R} \cdot \frac{\xi_2}{\sigma^2}, \tag{3.57}$$

$$\nu_{\mathsf{c}}^t \overset{\text{(i)}}{=} \frac{2}{R\,\mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\sigma^2 + \frac{1}{\mathsf{C}}\sum_{\mathsf{c}'} W_{\mathsf{rc}'}\psi_{\mathsf{c}'}^t} \overset{\text{(ii)}}{\geq} \frac{2}{R\,\mathsf{R}} \sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\sigma^2 + 2\xi_2} \overset{\text{(iii)}}{\geq} \frac{2}{R} \cdot \frac{\xi_1}{\sigma^2 + 2\xi_2}. \tag{3.58}$$

The labelled steps can be obtained as follows: (i) using the definition of ν_{c}^t and ϕ_{r}^t in (3.22) and (3.14), (ii) using (3.56) and $\xi_1 \leq \frac{1}{\mathsf{C}} \sum_{\mathsf{c}} W_{\mathsf{rc}} \leq \xi_2$, and (iii) using $\xi_1 \leq \frac{1}{\mathsf{R}} \sum_{\mathsf{r}} W_{\mathsf{rc}} \leq \xi_2$. Note that $W_{\mathsf{rc}} \geq 0$ for $\mathsf{r} \in [\mathsf{R}], \mathsf{c} \in [\mathsf{C}]$ and $\sigma^2 > 0$.

In the remainder of this proof, we obtain the upper bound on ψ_{c}^{t+1} given in (3.24). We do this by obtaining a lower bound on $\mathcal{E}(\tau_{\mathsf{c}}^t)$ since $\psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(\tau_{\mathsf{c}}^t)$. The result will first be proven for the $K \geq 8$ case. The other cases ($K = 1, 2, 4$) use similar arguments and will be discussed afterwards.

We first rewrite $\mathcal{E}(\tau)$ defined in (3.16) in terms of ν. Recalling from (3.22) that $\nu = \frac{1}{\tau \ln(KM)}$, we have

$$\mathcal{E}(\tau) = \mathbb{E}\left[\frac{\sum_{k=1}^{K} \Re[c_k] \cdot e^{\nu \ln(KM)\Re[c_k] + \sqrt{\nu \ln(KM)}\Re[U_1 \overline{c_k}]}}{\sum_{a=1}^{K} e^{\nu \ln(KM)\Re[c_a] + \sqrt{\nu \ln(KM)}\Re[U_1 \overline{c_a}]} + \sum_{j=2}^{M} \sum_{b=1}^{K} e^{\sqrt{\nu \ln(KM)}\Re[U_j \overline{c_b}]}}\right], \tag{3.59}$$

where $U_1, \ldots, U_M \overset{\text{i.i.d}}{\sim} \mathcal{CN}(0, 2)$, and $c_k = e^{\mathrm{j}2\pi k/K}$, for $1 \leq k \leq K$. Furthermore, we note that the set of PSK symbols $\{c_k\}_{k=1,\ldots,K}$ can equally be represented as $\{c_k\}_{k=i,\ldots,i+K-1}$ for any integer i, and we have $c_i = c_{i \bmod K}$. We now introduce some notation to simplify (3.59):

$$\cos_a = \cos(2\pi a/K) = \Re[c_a] \quad \text{for integer } a, \tag{3.60}$$

$$\sin_a = \sin(2\pi a/K) = \Im[c_a] \quad \text{for integer } a, \tag{3.61}$$

$$\tan_a = \tan(2\pi a/K) \quad \text{for integer } a, \tag{3.62}$$

$$U_j^R = \Re[U_j] \quad \text{for } j = 1, \ldots, M, \tag{3.63}$$

$$U_j^I = \Im[U_j] \quad \text{for } j = 1, \ldots, M, \tag{3.64}$$

$$\mu = \nu \ln(KM) = \frac{1}{\tau}. \tag{3.65}$$

Using the above notation, we have

$$\begin{aligned}
\mathcal{E}(\tau) &= \mathbb{E}\left[\frac{\sum_{k=1}^{K} \cos_k \; e^{\mu \cos_k + \sqrt{\mu}\Re[U_1 \overline{c_k}]}}{\sum_{a=1}^{K} e^{\mu \cos_a + \sqrt{\mu}\Re[U_1 \overline{c_a}]} + \sum_{j=2}^{M} \sum_{b=1}^{K} e^{\sqrt{\mu}\Re[U_j \overline{c_b}]}}\right] \\
&\overset{\text{(i)}}{=} \mathbb{E}\left[\frac{\sum_{k=-K/4+1}^{K/4} \cos_k \; e^{\mu \cos_k + \sqrt{\mu}\Re[U_1 \overline{c_k}]} - \cos_k \; e^{-\mu \cos_k - \sqrt{\mu}\Re[U_1 \overline{c_k}]}}{\sum_{a=-K/4+1}^{K/4} \left(e^{\mu \cos_a + \sqrt{\mu}\Re[U_1 \overline{c_a}]} + e^{-\mu \cos_a - \sqrt{\mu}\Re[U_1 \overline{c_a}]}\right) + \sum_{j=2}^{M} \sum_{b=1}^{K} e^{\sqrt{\mu}\Re[U_j \overline{c_b}]}}\right] \\
&\overset{\text{(ii)}}{=} \mathbb{E}\left[\frac{\sum_{k=-K/4+1}^{K/4} 2 \cos_k \cdot \sinh Y_k}{X + \sum_{a=-K/4+1}^{K/4} 2 \cosh Y_a}\right] = \mathbb{E}_{U_1} \mathbb{E}_X \left[\frac{\sum_{k=-K/4+1}^{K/4} 2 \cos_k \cdot \sinh Y_k}{X + \sum_{a=-K/4+1}^{K/4} 2 \cosh Y_a} \,\middle|\, U_1\right],
\end{aligned} \tag{3.66}$$

where (i) is obtained by using $c_a = -c_{(a+\frac{K}{2}) \bmod K}$ and $\cos_a = -\cos_{(a+\frac{K}{2}) \bmod K}$, noting that K is a multiple of 4, and (ii) is obtained with the following substitutions,

$$Y_k \;=\; \mu \cos_k + \sqrt{\mu}\Re\left[U_1 \overline{c_k}\right] \;=\; \mu \cos_k + \sqrt{\mu}\left[U_1^R \cos_k + U_1^I \sin_k\right], \quad k = -\frac{K}{4} + 1, \ldots, \frac{K}{4}, \tag{3.67}$$

$$X \;=\; \sum_{j=2}^{M}\sum_{b=1}^{K} e^{\sqrt{\mu}\Re[U_j\overline{c_b}]} \;=\; \sum_{j=2}^{M}\sum_{b=1}^{K} e^{\sqrt{\mu}\left[U_j^R\cos_b + U_j^I \sin_b\right]}. \tag{3.68}$$

From (3.55), we know that $-1 \le \mathcal{E}(\tau) \le 1$. Furthermore, the same arguments used to obtain (3.55) shows that the expectation over X in (3.66) is bounded as follows,

$$-1 \le \mathbb{E}_X\left[\frac{\sum_{k=-K/4+1}^{K/4} 2\cos_k \cdot \sinh Y_k}{X + \sum_{a=-K/4+1}^{K/4} 2\cosh Y_a}\,\middle|\, U_1\right] \le 1. \tag{3.69}$$

To lower bound $\mathcal{E}(\tau)$, we first identify when the expression inside the inner expectation in (3.66) is non-negative. Observe that $\cos_k$, X, and $\cosh Y_a$ are all non-negative for the values of a and k being considered, with $\cos_k = 0$ when $k = \frac{K}{4}$. Furthermore, from the definition of Y_k in (3.67), for $k \in \{-K/4+1, \ldots, K/4-1\}$, we have $\sinh Y_k \ge 0$ if and only if

$$U_1^R\cos_k + U_1^I \sin_k \ge -\sqrt{\mu}\cos_k\,. \tag{3.70}$$

It can be easily verified that a sufficient condition for (3.70) to hold for $k \in \{-K/4+1, \ldots, K/4-1\}$ is that both U_1^R and U_1^I are greater than or equal to $-\sqrt{\mu}/(1+\tan(\frac{\pi}{2}-\frac{2\pi}{K}))$.

Using this knowledge, we can split the expectation over U_1 in (3.66) into integrals over four regions such that in at least one of the regions the integrand is non-negative. For any $\underline{u} < 0$, we have

$$\begin{aligned}
\mathcal{E}(\tau) &= \int_{\underline{u}}^{\infty}\int_{\underline{u}}^{\infty} p(u^R)\,p(u^I)\,\mathbb{E}_X\left[\frac{\sum_{k=-K/4+1}^{K/4} 2\cos_k\cdot\sinh Y_k}{X+\sum_{a=-K/4+1}^{K/4} 2\cosh Y_a}\,\middle|\, U_1^R = u^R, U_1^I = u^I\right] du^R\,du^I \\
&\quad + \int_{-\infty}^{\underline{u}}\int_{-\infty}^{\underline{u}} p(u^R)\,p(u^I)\,\mathbb{E}_X[\ldots]du^R\,du^I + \int_{\underline{u}}^{\infty}\int_{-\infty}^{\underline{u}} p(u^R)\,p(u^I)\,\mathbb{E}_X[\ldots]du^R\,du^I \\
&\quad + \int_{-\infty}^{\underline{u}}\int_{\underline{u}}^{\infty} p(u^R)\,p(u^I)\,\mathbb{E}_X[\ldots]du^R\,du^I \\
&\overset{\text{(i)}}{\ge} \int_{\underline{u}}^{\infty}\int_{\underline{u}}^{\infty} p(u^R)\,p(u^I)\,\mathbb{E}_X[\ldots]du^R\,du^I + \int_{-\infty}^{\underline{u}}\int_{-\infty}^{\underline{u}} p(u^R)\,p(u^I)\,(-1)\,du^R\,du^I \\
&\quad + \int_{\underline{u}}^{\infty}\int_{-\infty}^{\underline{u}} p(u^R)\,p(u^I)\,(-1)\;du^R\,du^I + \int_{-\infty}^{\underline{u}}\int_{\underline{u}}^{\infty} p(u^R)\,p(u^I)\,(-1)\,du^R\,du^I \\
&\overset{\text{(ii)}}{=} I_1 - Q(|\underline{u}|)^2 - 2[1-Q(|\underline{u}|)]\,Q(|\underline{u}|) \\
&\ge I_1 - 2Q(|\underline{u}|),
\end{aligned} \tag{3.71}$$

where I_1 is the integral over the first region. Step (i) is obtained using (3.69), and step (ii) using $U_1^R, U_1^I \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$, and defining $Q(x) = \int_x^{\infty}\frac{1}{\sqrt{2\pi}}e^{-z^2/2}\,dz$ to be the upper tail probability of the standard Gaussian distribution.

From (3.70) and the discussion surrounding it, when $\underline{u} = -\sqrt{\mu}/(1+\tan(\frac{\pi}{2}-\frac{2\pi}{K})) = -\sqrt{\mu}/(1+\cot(\frac{2\pi}{K}))$, the integrand of I_1 is non-negative, and we therefore have the following lower bound

for $\mathcal{E}(\tau)$:

$$\mathcal{E}(\tau) \geq -2Q(|\underline{u}|) \geq -\frac{2(1+\cot(\frac{2\pi}{K}))}{\sqrt{2\pi\nu\ln(KM)}}\,(KM)^{-\frac{\nu}{2(1+\cot(\frac{2\pi}{K}))^2}}, \tag{3.72}$$

where the second inequality is obtained using the bound $Q(x) \leq \frac{1}{x\sqrt{2\pi}}e^{-x^2/2}$ for $x > 0$.

The lower bound of $\mathcal{E}(\tau)$ in (3.72) applies for all $\nu > 0$. We now show that for $\nu > 2$, one can obtain a better lower bound which shows that $\mathcal{E}(\tau)$ approaches the upper bound of 1 with growing M. We do this by using a different choice for $\underline{u}$ to split the expectation over U_1 in (3.66) into integrals over four different regions. Let

$$\underline{u} = \frac{-\alpha(\frac{\nu}{2}-1)}{\nu}\frac{\sqrt{\mu}}{1+\cot(\frac{2\pi}{K})}\,. \tag{3.73}$$

Then, for any $\alpha \in (0,1)$ and $\nu > 2$, when $U_1^R \geq \underline{u}$ and $U_1^I \geq \underline{u}$ we have

$$U_1^R\cos_k + U_1^I\sin_k \geq -\sqrt{\mu}\,\cos_k \cdot \frac{\alpha(\frac{\nu}{2}-1)}{\nu}\cdot\frac{1+\tan_k}{1+\tan(\frac{\pi}{2}-\frac{2\pi}{K})} \geq -\sqrt{\mu}\,\cos_k\,, \tag{3.74}$$

for $\frac{-K}{4}+1 \leq k \leq \frac{K}{4}-1$. Thus, under these conditions, (3.70) holds and the integrand of I_1 in (3.71) is non-negative. In the following lemma, we obtain a stronger lower bound on I_1 for $\nu > 2$.

Lemma 3.6.1. *When* $\underline{u} = \frac{-\alpha(\frac{\nu}{2}-1)}{\nu}\frac{\sqrt{\mu}}{1+\cot(\frac{2\pi}{K})}$ *for any* $\alpha \in (0,1)$, $\nu > 2$ *and* $K \geq 4$, *the* I_1 *term in* (3.71) *can be lower bounded as follows:*

$$I_1 \geq 1 - 3Q(|\underline{u}|) - 2(KM)^{-2z} - (KM)^{1+\frac{\nu}{2}-z} - (K-2)(KM)^{-(z-z^\star)}, \tag{3.75}$$

where

$$z = \nu - \frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(\frac{2\pi}{K})}, \tag{3.76}$$

$$z^\star = z\cos\Big(\frac{2\pi}{K}\Big) + (\nu - z)\sin\Big(\frac{2\pi}{K}\Big). \tag{3.77}$$

The proof is given in Section 3.6.3.

Applying the result of Lemma 3.6.1 into (3.71), we have, for any $\alpha \in (0,1)$ and $\nu > 2$,

$$\begin{aligned}
\mathcal{E}(\tau) &\geq 1 - 5Q(|\underline{u}|) - 2(KM)^{-2z} - (KM)^{1+\frac{\nu}{2}-z} - (K-2)(KM)^{-(z-z^\star)} \\
&\overset{\text{(i)}}{=} 1 - 5Q(|\underline{u}|) - 2(KM)^{-2\left(\nu - \frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(\frac{2\pi}{K})}\right)} - (KM)^{-\left(1-\frac{\alpha}{1+\cot(\frac{2\pi}{K})}\right)(\frac{\nu}{2}-1)} \\
&\qquad - (K-2)(KM)^{-\left(\nu-\frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(\frac{2\pi}{K})}\right)(1-\cos(\frac{2\pi}{K}))+\frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(\frac{2\pi}{K})}\cdot\sin(\frac{2\pi}{K})} \\
&\geq 1 - 5Q(|\underline{u}|) - (KM)^{-\left(1-\frac{\alpha}{1+\cot(\frac{2\pi}{K})}\right)(\frac{\nu}{2}-1)}
\end{aligned}$$

$$
\begin{aligned}
&\quad - K(KM)^{-\left(\nu-\frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(\frac{2\pi}{K})}\right)\left(1-\cos(\frac{2\pi}{K})\right)+\frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(\frac{2\pi}{K})}\cdot\sin(\frac{2\pi}{K})} \\
&\overset{\text{(ii)}}{\geq} 1-\frac{5\sqrt{\nu}(1+\cot(\frac{2\pi}{K}))(KM)^{-\frac{\alpha^2(\frac{\nu}{2}-1)^2}{2\nu(1+\cot(\frac{2\pi}{K}))^2}}}{(\nu/2-1)\alpha\sqrt{2\pi\ln(KM)}}-(KM)^{-\left(1-\frac{\alpha}{1+\cot(\frac{2\pi}{K})}\right)(\frac{\nu}{2}-1)} \\
&\quad - K(KM)^{-\left[1-\cos(\frac{2\pi}{K})\right]\cdot\left[\nu\left[1-\frac{\alpha}{2}\left(1+\frac{1}{\sin(\frac{2\pi}{K})+\cos(\frac{2\pi}{K})}\right)\right]+\alpha\left(1+\frac{1}{\sin(\frac{2\pi}{K})+\cos(\frac{2\pi}{K})}\right)\right]},
\end{aligned}
\tag{3.78}
$$

where (i) is obtained using the substitutions (3.76) and (3.77), and (ii) from using the bound $Q(x) \leq \frac{1}{x\sqrt{2\pi}}e^{-x^2/2}$ for $x > 0$ (noting that $\underline{u} < 0$ for $\nu > 2$), and rearranging the exponent of the last term as follows. Dropping the arguments of $\sin(\cdot)$, $\cos(\cdot)$ and $\cot(\cdot)$ for brevity, we have

$$
\begin{aligned}
&-\left(\nu-\frac{\alpha(\frac{\nu}{2}-1)}{1+\cot}\right)(1-\cos)+\frac{\alpha(\frac{\nu}{2}-1)}{1+\cot}\cdot\sin \\
&=-(1-\cos)\left[\nu-\frac{\alpha(\nu/2-1)}{1+\cot}-\frac{\alpha(\nu/2-1)}{1+\cot}\frac{\sin}{1-\cos}\right] \\
&=-(1-\cos)\left[\nu\Big(1-\frac{\alpha/2}{(1+\cot)}\Big(1+\frac{\sin}{1-\cos}\Big)\Big)+\frac{\alpha}{(1+\cot)}\Big(1+\frac{\sin}{1-\cos}\Big)\right] \\
&=-(1-\cos)\left[\nu\Big(1-\frac{\alpha}{2}\Big(1+\frac{1}{\sin+\cos}\Big)\Big)+\alpha\Big(1+\frac{1}{\sin+\cos}\Big)\right].
\end{aligned}
$$

For any $\delta \in (0, \frac{1}{2})$, we consider the case $\nu > 2+\delta$ and choose $\alpha = 1-\delta$. With this choice, the exponent of the last term in (3.78) can be bounded as follows:

$$
\begin{aligned}
&\nu\big[1-\frac{\alpha}{2}\big(1+(\sin(2\pi/K)+\cos(2\pi/K))^{-1}\big)\big]+\alpha\big(1+(\sin(2\pi/K)+\cos(2\pi/K))^{-1}\big) \\
&>2+\delta\left[1-\frac{\alpha}{2}\left(1+(\sin(2\pi/K)+\cos(2\pi/K))^{-1}\right)\right] \\
&\geq 2+\delta(1-\alpha)=2+\delta^2,
\end{aligned}
\tag{3.79}
$$

where the second inequality holds because $\sin(\frac{2\pi}{K})+\cos(\frac{2\pi}{K}) \geq 1$ for $K \geq 8$.

Using (3.79) in (3.78) along with $\frac{\nu}{2}-1 > \frac{\delta}{2}$ and $\alpha = 1-\delta$, we obtain

$$
\begin{aligned}
&\mathcal{E}(\tau) \\
&\geq 1-\frac{10\sqrt{\nu}\,(1+\cot(\frac{2\pi}{K}))(KM)^{-\frac{(1-\delta)^2\delta^2}{8\nu(1+\cot(\frac{2\pi}{K}))^2}}}{(1-\delta)\,\delta\sqrt{2\pi\ln(KM)}}-(KM)^{-\frac{\delta(\delta+\cot(\frac{2\pi}{K}))}{2(1+\cot(\frac{2\pi}{K}))}}-K(KM)^{-(2+\delta^2)(1-\cos(\frac{2\pi}{K}))} \\
&\geq 1-\frac{(1+\cot(\frac{2\pi}{K}))(KM)^{-\frac{\kappa\delta^2}{(1+\cot(\frac{2\pi}{K}))^2}}}{\delta\sqrt{\ln(KM)}}-K(KM)^{-2(2+\delta^2)\sin^2(\frac{\pi}{K})},
\end{aligned}
\tag{3.80}
$$

where $\kappa < \frac{(1-\delta)^2}{8\nu}$ is a suitably chosen universal positive constant (for M sufficiently large). For the second inequality we used $1-\delta > \frac{1}{2}$ and the fact that ν can be upper bounded by a positive

constant (3.57). Comparing the exponents of the last two terms of (3.80), using $\kappa < \frac{(1-\delta)^2}{8\nu}$, $\delta \in (0, \frac{1}{2})$ and $\nu > 2$, we have

$$\begin{aligned}\frac{\kappa\delta^2}{(1+\cot(\frac{2\pi}{K}))^2} &< \frac{(1-\delta)^2\delta^2}{8\nu(1+\cot(\frac{2\pi}{K}))^2} < \frac{1}{256\,(1+\cos_1/\sin_1)^2} = \frac{1-\cos_1^2}{256\,(\sin_1+\cos_1)^2} \\ &= \frac{(1-\cos_1)(1+\cos_1)}{256\,(\sin_1+\cos_1)^2} < \frac{2(1-\cos_1)}{256} < \frac{2(2+\delta^2)\sin^2(\pi/K)}{256},\end{aligned}$$

where we have used the notation $\cos_1 = \cos(2\pi/K)$ and $\sin_1 = \sin(2\pi/K)$. The second last inequality above is obtained using $\cos_1 < 1$, and $(\sin_1 + \cos_1)^2 = [\sqrt{2}\sin(\frac{\pi}{4} + \frac{2\pi}{K})]^2 \geq 1$ for $K \geq 8$. Therefore, the exponent of third term is more than 256 times larger than that of second term (in absolute value).

Summing up, we now have two lower bounds on $\mathcal{E}(\tau)$ for different values of ν. When $\nu \leq 2+\delta$ we have (3.72), and when $\nu > 2 + \delta$ we have (3.80). By applying them to the state evolution equation $\psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(\tau_{\mathsf{c}}^t)$, we obtain the required result for the $K \geq 8$ case. The results for the other cases ($K = 1, 2, 4$) use similar arguments and are explained below.

***K* = 1:** There is only one constellation symbol $c_0 = 1$ which has no imaginary part. Therefore, all imaginary parts in the expression of $\mathcal{E}(\tau)$ disappear as shown below.

$$\mathcal{E}(\tau) = \mathbb{E}\left[\frac{e^{\nu\ln(M)+\sqrt{\nu\ln(M)}U_1^R}}{e^{\nu\ln(M)+\sqrt{\nu\ln(M)}U_1^R} + \sum_{j=2}^{M} e^{\sqrt{\nu\ln(M)}U_j^R}}\right],$$

where $U_1^R, \ldots, U_M^R \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. The above expression is the same as that given in [57, Eq. (A.2)]. Therefore, we can obtain the result from following the steps in [57, App. A].

***K* = 2:** In this case there are two constellation symbols $c_0 = 1$ and $c_1 = -1$, which are both real. We follow steps similar to those used to obtain (3.71) in the $K \geq 8$ case to obtain the following lower bound on $\mathcal{E}(\tau)$ for any $\underline{u} \leq 0$

$$\begin{aligned}\mathcal{E}(\tau) &= \mathbb{E}\left[\frac{2\sinh(\mu+\sqrt{\mu}\,U_1^R)}{2\cosh(\mu+\sqrt{\mu}\,U_1^R) + \sum_{j=2}^{M}(e^{\sqrt{\mu}\,U_j^R} + e^{-\sqrt{\mu}\,U_j^R})}\right] \\ &= \mathbb{E}_{U_1^R}\mathbb{E}_X\left[\frac{2\sinh(\mu+\sqrt{\mu}\,U_1^R)}{2\cosh(\mu+\sqrt{\mu}\,U_1^R)+X}\,\middle|\,U_1^R\right] \\ &= \int_{\underline{u}}^{\infty} p(u)\,\mathbb{E}_X\left[\ldots \mid U_1^R = u\right]du + \int_{-\infty}^{\underline{u}} p(u)\,\mathbb{E}_X\left[\ldots \mid U_1^R = u\right]du \\ &\overset{\text{(i)}}{\geq} \int_{\underline{u}}^{\infty} p(u)\,\mathbb{E}_X\left[\ldots \mid U_1^R = u\right]du + \int_{-\infty}^{\underline{u}} p(u)\,(-1)du \overset{\text{(ii)}}{=} I_1 - Q(|\underline{u}|),\end{aligned} \tag{3.81}$$

where $\mu = \nu\ln(2M)$, $X = \sum_{j=2}^{M}(e^{\sqrt{\mu}\,U_j^R} + e^{-\sqrt{\mu}\,U_j^R})$. Step (i) is obtained using $|\frac{2\sinh(x)}{2\cosh(x)+X}| \leq 1$ for any x, and step (ii) holds since $\underline{u} \leq 0$.

Similar to what was done in (3.72) for the $K \geq 8$ case, we choose $\underline{u} = -\sqrt{\mu} = -\sqrt{\nu\ln(2M)}$

so that the integrand of I_1 is non-negative, and obtain the following lower bound on $\mathcal{E}(\tau)$ for all $\nu > 0$.

$$\mathcal{E}(\tau) \overset{\text{(i)}}{\geq} -Q\left(\sqrt{\nu \ln(2M)}\right) \overset{\text{(ii)}}{\geq} -\frac{(2M)^{-\frac{\nu}{2}}}{\sqrt{2\pi\nu\ln(2M)}}, \tag{3.82}$$

where the labelled inequalities are obtained as follows: (i) $I_1 \geq 0$ when $\underline{u} = -\sqrt{\mu}$, and (ii) using the bound on the tail probability of a standard Gaussian.

The lower bound on $\mathcal{E}(\tau)$ in (3.82) applies for all values of $\nu > 0$. To obtain a better bound for $\nu > 2$ which shows that $\mathcal{E}(\tau)$ approaches 1 with growing M, we choose

$$\underline{u} = \frac{-\alpha(\frac{\nu}{2}-1)}{\nu}\sqrt{\mu} = -\alpha\Big(\frac{\nu}{2}-1\Big)\sqrt{\frac{\ln(2M)}{\nu}} \tag{3.83}$$

for any $\alpha \in (0,1)$, and then using this choice of $\underline{u}$ in (3.81) to obtain a lower bound for I_1. We obtain the following lower bound on I_1 by following similar steps to the proof of Lemma 3.6.1 given in Section 3.6.3. Recall that Lemma 3.6.1 obtains a lower bound on I_1 for $\nu > 2$ in the $K \geq 4$ case.

$$I_1 \geq 1 - Q(|\underline{u}|) - 2\,(2M)^{-((2-\alpha)\nu+2\alpha)} - (2M)^{-(1-\alpha)(\frac{\nu}{2}-1)}. \tag{3.84}$$

Using (3.84) in (3.81), we have the following lower bound on $\mathcal{E}(\tau)$ for $\nu > 2$ and $\underline{u}$ taking the value in (3.83).

$$\mathcal{E}(\tau) \geq 1 - \frac{2\sqrt{\nu}(2M)^{-\frac{\alpha^2(\frac{\nu}{2}-1)^2}{2\nu}}}{\alpha(\frac{\nu}{2}-1)\sqrt{2\pi\ln(2M)}} - 2\,(2M)^{-((2-\alpha)\nu+2\alpha)} - (2M)^{-(1-\alpha)(\frac{\nu}{2}-1)}, \tag{3.85}$$

where we used the bound $Q(x) \leq \frac{1}{x\sqrt{2\pi}}e^{-x^2/2}$ for $x > 0$.

For any $\delta \in (0, \frac{1}{2})$, we consider the case $\nu > 2+\delta$ and choose $\alpha = 1-\delta$. Plugging these into (3.85), we have

$$\mathcal{E}(\tau) \geq 1 - \frac{4\sqrt{\nu}(2M)^{-\frac{\delta^2(1-\delta)^2}{8\nu}}}{\delta(1-\delta)\sqrt{2\pi\ln(2M)}} - 2\,(2M)^{-(4+\delta+\delta^2)} - (2M)^{-\frac{\delta^2}{2}} \geq 1 - \frac{(2M)^{-\kappa_2\delta^2}}{\delta\sqrt{\ln(2M)}}, \tag{3.86}$$

for sufficiently large M and a suitably chosen universal positive constant κ_2. The second inequality is obtained by noting that $1-\delta > \frac{1}{2}$ and that ν can be upper bounded by a positive constant (3.57).

By applying the two lower bounds (3.82) and (3.86) to the state evolution equation $\psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(\tau_{\mathsf{c}}^t)$ for the $\nu \leq 2+\delta$ and $\nu > 2+\delta$ case respectively, we obtain the required result for $K = 2$.

***K* = 4:** The arguments for this case are essentially the same as that for the $K \geq 8$ case, noting that $\tan(\frac{\pi}{2} - \frac{2\pi}{K}) = \cot(\frac{2\pi}{K}) = 0$ and $\cos(\frac{2\pi}{K}) = 0$ for $K = 4$.

By following the same arguments from (3.59)–(3.72), we obtain the following lower bound for $\mathcal{E}(\tau)$ for any $\nu > 0$,

$$\mathcal{E}(\tau) \geq \frac{-2(4M)^{-\frac{\nu}{2}}}{\sqrt{2\pi\nu \ln(4M)}}. \tag{3.87}$$

To obtain the improved lower bound for the $\nu > 2$ case, we follow the same arguments from (3.74) to (3.80) and arrive at the following. For any $\delta \in (0, \frac{1}{2})$ and $\nu > 2 + \delta$, we have

$$\mathcal{E}(\tau) \geq 1 - \frac{10\sqrt{\nu}(4M)^{-\frac{(1-\delta)^2\delta^2}{8\nu}}}{(1-\delta)\delta\sqrt{2\pi \ln(4M)}} - (4M)^{-\frac{\delta^2}{2}} - 4(4M)^{-2-\delta^2} \geq 1 - \frac{(4M)^{-\kappa_3\delta^2}}{\delta\sqrt{\ln(4M)}}, \tag{3.88}$$

for sufficiently large M and a suitably chosen universal positive constant κ_3. The last inequality is obtained by noting that $1 - \delta > \frac{1}{2}$ and that ν can be upper bounded by a positive constant (3.57).

By applying the two lower bounds (3.87) and (3.88) to the state evolution equation $\psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(\tau_{\mathsf{c}}^t)$ for the $\nu \leq 2 + \delta$ and $\nu > 2 + \delta$ case respectively, we obtain the required result for $K = 4$.

3.6.3 Proof of Lemma 3.6.1

From (3.71), we have

$$I_1 = \int_{\underline{u}}^{\infty} \int_{\underline{u}}^{\infty} p(u^R)\, p(u^I)\, \mathbb{E}_X \left[\frac{\sum_{k=-K/4+1}^{K/4} 2\cos_k \cdot \sinh Y_k}{X + \sum_{a=-K/4+1}^{K/4} 2\cosh Y_a} \,\middle|\, U_1^R = u^R, U_1^I = u^I \right] du^R\, du^I, \tag{3.89}$$

where $\cos_k$, Y_k for $k \in \{-K/4+1, \ldots, K/4\}$ and X are defined in (3.60), (3.67) and (3.68) respectively, $K \geq 4$, and $p(u^R)$, $p(u^I)$ are standard Gaussian densities. We aim to lower bound I_1 with $\underline{u}$ taking the value

$$\underline{u} = \frac{-\alpha(\frac{\nu}{2} - 1)}{\nu} \frac{\sqrt{\mu}}{1 + \cot(\frac{2\pi}{K})} \tag{3.90}$$

for any $\alpha \in (0, 1)$ and $\nu > 2$. Recall from the arguments around equation (3.73) and (3.74) that the integrand of I_1 is non-negative for $\nu > 2$ with this choice of $\underline{u}$.

Using Jensen's inequality for the expectation in (3.89), we obtain

$$\begin{aligned} I_1 &\geq \int_{\underline{u}}^{\infty} \int_{\underline{u}}^{\infty} p(u^R)\, p(u^I)\, \frac{\sum_k 2\cos_k \cdot \sinh Y_k}{\mathbb{E}X + \sum_a 2\cosh Y_a}\, du^R\, du^I \\ &\overset{\text{(i)}}{\geq} \int_{\underline{u}}^{\infty} \int_{\underline{u}}^{\infty} p(u^R)\, p(u^I)\, \frac{\sum_k 2\cos_k \cdot \sinh Y_k}{(KM)^{1+\frac{\nu}{2}} + \sum_a 2\cosh Y_a}\, du^R\, du^I \\ &\overset{\text{(ii)}}{\geq} \int_{\underline{u}}^{\infty} \int_{\underline{u}}^{\infty} p(u^R)\, p(u^I)\, \frac{2\sinh Y_0}{(KM)^{1+\frac{\nu}{2}} + \sum_a 2\cosh Y_a}\, du^R\, du^I, \end{aligned} \tag{3.91}$$

where in the above, the summations over a and k go from $-K/4+1$ to $K/4$ and are omitted for

brevity. Inequality (i) is obtained as follows using the definition of X in (3.68) and the moment generating function of a standard Gaussian:

$$\mathbb{E}X = \sum_{j=2}^{M}\sum_{b=1}^{K}\mathbb{E}\big[e^{\sqrt{\mu}\cos_b U_j^R}\big]\mathbb{E}\big[e^{\sqrt{\mu}\sin_b U_j^I}\big] = \sum_{j=2}^{M}\sum_{b=1}^{K} e^{\frac{\mu}{2}(\cos_k^2+\sin_k^2)} = \sum_{j=2}^{M}\sum_{b=1}^{K}(KM)^{\frac{\nu}{2}} < (KM)^{1+\frac{\nu}{2}},$$

where we have used $\mu = \nu\ln(KM)$. Inequality (ii) in (3.91) is obtained by taking only the $k=0$ term in the numerator's summation, recalling that $\cos_k$ and $\sinh Y_k$ for $k \in \{-K/4+1,\ldots,K/4\}$ are all non-negative when $U_1^R \geq \underline{u}$ and $U_1^I \geq \underline{u}$.

Next, we further lower bound I_1 using the fact that the term $\frac{2\sinh Y_0}{((KM)^{1+\frac{\nu}{2}}+\sum_a 2\cosh Y_a)}$ in (3.91) is a strictly increasing function of U_1^R. (Recall from (3.67) that Y_k is a function of U_1^R for each k.) To see that the term is increasing in U_1^R, we write

$$f \coloneqq \frac{2\sinh Y_0}{(KM)^{1+\frac{\nu}{2}}+\sum_a 2\cosh Y_a} = \frac{f_{\text{num}}}{f_{\text{den}}}, \tag{3.92}$$

and show that $\partial f/\partial U_1^R > 0$. Since

$$\frac{\partial f}{\partial U_1^R} = \Big(\frac{\partial f_{\text{num}}}{\partial U_1^R}\cdot f_{\text{den}} - f_{\text{num}}\cdot\frac{\partial f_{\text{den}}}{\partial U_1^R}\Big)\Big/\Big(f_{\text{den}}^2\Big), \tag{3.93}$$

we show below that $\frac{\partial f_{\text{num}}}{\partial U_1^R}\cdot f_{\text{dem}} - f_{\text{num}}\cdot\frac{\partial f_{\text{dem}}}{\partial U_1^R} > 0$. Indeed,

$$\begin{aligned}
&\frac{\partial f_{\text{num}}}{\partial U_1^R}\cdot f_{\text{den}} - f_{\text{num}}\cdot\frac{\partial f_{\text{den}}}{\partial U_1^R}\\
&= (2\sqrt{\mu}\cos_0\cosh Y_0)((KM)^{1+\frac{\nu}{2}}+\sum_a 2\cosh Y_a) - (2\sinh Y_0)(\sum_a 2\sqrt{\mu}\cos_a\sinh Y_a)\\
&= 2\sqrt{\mu}\cosh Y_0\cdot[(KM)^{1+\frac{\nu}{2}} + 2\sum_a(\cosh Y_a - \frac{\sinh Y_0}{\cosh Y_0}\cos_a\sinh Y_a)]\\
&= 2\sqrt{\mu}\cosh Y_0\cdot[(KM)^{1+\frac{\nu}{2}} + 2\sum_a\cosh Y_a\cdot(1-\tanh(Y_0)\tanh(Y_a)\cos_a)] > 0.
\end{aligned} \tag{3.94}$$

In the above, the summations over a go from $-K/4+1$ to $K/4$. The inequality in (3.94) holds because $0 \leq \cos_a \leq 1$ for $a \in \{-K/4+1,\ldots,K/4\}$, and $|\tanh(x)| < 1$ for all $x \in \mathbb{R}$.

Using the fact that the integrand in (3.91) is strictly increasing in U_1^R, we further bound I_1 from below using the minimum value of U_1^R in the range, i.e., $U_1^R = \underline{u}$ where $\underline{u}$ is given by (3.90). Using the expressions for Y_0 and Y_a from (3.67), and the substitution

$$z = \frac{(\mu+\sqrt{\mu}\,\underline{u})}{\ln(KM)} = \nu - \frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(\frac{2\pi}{K})}, \tag{3.95}$$

the RHS of (3.91) is lower bounded as

$$
\begin{aligned}
I_1 &\geq \int_{\underline{u}}^{\infty}\int_{\underline{u}}^{\infty} p(u^R)\,p(u^I)\,\frac{2\sinh[z\ln(KM)]}{(KM)^{1+\frac{\nu}{2}}+\sum_a 2\cosh[z\ln(KM)\cos_a+\sqrt{\mu}u^I\sin_a]}\,du^R\,du^I \\
&\overset{\text{(i)}}{=} Q(\underline{u})\int_{\underline{u}}^{\infty} p(u)\,\frac{2\sinh[z\ln(KM)]}{(KM)^{1+\frac{\nu}{2}}+\sum_a 2\cosh[z\ln(KM)\cos_a+\sqrt{\mu}u\sin_a]}\,du \\
&\geq Q(\underline{u})\int_{\underline{u}}^{-\underline{u}} p(u)\,\frac{2\sinh[z\ln(KM)]}{(KM)^{1+\frac{\nu}{2}}+\sum_a 2\cosh[z\ln(KM)\cos_a+\sqrt{\mu}u\sin_a]}\,du \\
&\overset{\text{(ii)}}{=} Q(\underline{u})\int_{\underline{u}}^{-\underline{u}} p(u)\,\frac{2\sinh[z\ln(KM)]}{(KM)^{1+\frac{\nu}{2}}+(1+\Delta(u))\cdot 2\cosh[z\ln(KM)]}\,du \\
&= Q(\underline{u})\int_{\underline{u}}^{-\underline{u}} p(u)\,\frac{1-(KM)^{-2z}}{1+(KM)^{-2z}+(KM)^{1+\frac{\nu}{2}-z}+\Delta(u)(1+(KM)^{-2z})}\,du \\
&\overset{\text{(iii)}}{\geq} Q(\underline{u})\int_{\underline{u}}^{-\underline{u}} p(u)\,[1-(KM)^{-2z}][1-(KM)^{-2z}-(KM)^{1+\frac{\nu}{2}-z}-\Delta(u)(1+(KM)^{-2z})]\,du \\
&\geq Q(\underline{u})\int_{\underline{u}}^{-\underline{u}} p(u)\,[1-2(KM)^{-2z}-(KM)^{1+\frac{\nu}{2}-z}-\Delta(u)(1+(KM)^{-2z})]\,du \\
&\overset{\text{(iv)}}{\geq} Q(\underline{u})\int_{\underline{u}}^{-\underline{u}} p(u)\,[1-2(KM)^{-2z}-(KM)^{1+\frac{\nu}{2}-z}-(K-2)(KM)^{-(z-z^\star)}]\,du \\
&\overset{\text{(v)}}{=} [1-Q(|\underline{u}|)]\,[1-2Q(|\underline{u}|)]\,[1-2(KM)^{-2z}-(KM)^{1+\frac{\nu}{2}-z}-(K-2)(KM)^{-(z-z^\star)}] \\
&\geq 1-3Q(|\underline{u}|)-2(KM)^{-2z}-(KM)^{1+\frac{\nu}{2}-z}-(K-2)(KM)^{-(z-z^\star)}. \qquad (3.96)
\end{aligned}
$$

The labelled steps are obtained as follows: (i) recalling that $U_1^R \sim \mathcal{N}(0,1)$ and dropping the superscript on u^I for brevity; (ii) introducing the substitution

$$\Delta(u) = \sum_{a\in\{-K/4+1,\ldots,K/4\}\setminus 0} \frac{\cosh[z\ln(KM)\cos_a+\sqrt{\mu}u\sin_a]}{\cosh[z\ln(KM)]}; \qquad (3.97)$$

(iii) using $\frac{1}{1+x}\geq 1-x$ for $x\geq 0$, noting that $\Delta(u)>0$; (iv) from the upper bound on $\Delta(u)$ for $|u|\leq -\underline{u}$ shown below in (3.98); and (v) from recalling that $p(u)$ is the density of a standard Gaussian and noting that $\underline{u}<0$.

To complete the proof of the lemma, it remains to show the upper bound on $\Delta(u)$ that is used in step (iv) of (3.96). Using the definition of $\Delta(u)$ in (3.97), for $|u|\leq -\underline{u}$ we have

$$
\begin{aligned}
\Delta(u) &\overset{\text{(i)}}{\leq} \sum_{a\in\{-K/4+1,\ldots,K/4\}\setminus 0} \frac{\cosh[z\ln(KM)\cos_a+\sqrt{\mu}|u||\sin_a|]}{\cosh[z\ln(KM)]} \\
&= \sum_{a\in\{-K/4+1,\ldots,K/4\}\setminus 0} \frac{\cosh[z\ln(KM)\cos_a+\frac{\alpha(\frac{\nu}{2}-1)}{1+\cot(2\pi/K)}\ln(KM)|\sin_a|]}{\cosh[z\ln(KM)]} \\
&\overset{\text{(ii)}}{\leq} \Big(\frac{K}{2}-1\Big)\frac{\max_{a\in\{1,\ldots,K/4\}}\cosh[z\ln(KM)\cos_a+(\nu-z)\ln(KM)\sin_a]}{\cosh[z\ln(KM)]}
\end{aligned}
$$

$$\overset{\text{(iii)}}{=} \Big(\frac{K}{2}-1\Big)\frac{\cosh[z^\star \ln(KM)]}{\cosh[z\ln(KM)]} = \Big(\frac{K}{2}-1\Big)(KM)^{-(z-z^\star)}\frac{1+(KM)^{-2z^\star}}{1+(KM)^{-2z}}$$

$$\overset{\text{(iv)}}{\leq} \frac{(K-2)(KM)^{-(z-z^\star)}}{1+(KM)^{-2z}}, \tag{3.98}$$

where the labelled steps are obtained as follows: (i) for $u \geq \underline{u}$, we know

$$z\ln(KM)\cos_a + \sqrt{\mu}u\sin_a \geq 0$$

and that $\cosh(x)$ is an increasing function for $x \geq 0$; (ii) using (3.95) and noting that taking the maximum over $a \in \{-K/4+1,\ldots,K/4\}\backslash 0$ is the same as taking the maximum over $a \in \{1,\ldots,K/4\}$ since $\cos_a = \cos_{-a}$ and $|\sin_a| = |\sin_{-a}|$; (iii) holds because the maximum is achieved with $a=1$ (shown below) and defining $z^\star$ as

$$z^\star = z\cos\Big(\frac{2\pi}{K}\Big) + (\nu - z)\sin\Big(\frac{2\pi}{K}\Big), \tag{3.99}$$

where z is defined in (3.95); and (iv) using $1+(KM)^{-2z^\star} \leq 2$ since $z^\star \geq 0$.

In order to show that $a=1$ achieves the maximum of

$$\begin{aligned} f_1(a) &:= \cosh[z\ln(KM)\cos_a + (\nu - z)\ln(KM)\sin_a] \\ &= \cosh\left[\nu\ln(KM)\left(\left(1 - \frac{\alpha(\frac{1}{2}-\frac{1}{\nu})}{1+\cot(2\pi/K)}\right)\cos_a + \frac{\alpha(\frac{1}{2}-\frac{1}{\nu})}{1+\cot(2\pi/K)}\sin_a\right)\right] \end{aligned} \tag{3.100}$$

for $a \in \{1,\ldots,K/4\}$, we show that $f_1'(a) < 0$ for all a in this region. Since $\cosh(x)$ is increasing function for $x \geq 0$ and the argument of the cosh term in (3.100) is non-negative, we only need to show that $f_2'(a) < 0$ for all $a \in \{1,\ldots,K/4\}$, where

$$f_2(a) := \left(1 - \frac{\alpha(\frac{1}{2}-\frac{1}{\nu})}{1+\cot(2\pi/K)}\right)\cos_a + \frac{\alpha(\frac{1}{2}-\frac{1}{\nu})}{1+\cot(2\pi/K)}\sin_a\,. \tag{3.101}$$

For $\alpha \in (0,1)$ and $\nu > 2$, we have

$$\begin{aligned} f_2'(a) &= \frac{2\pi}{K}\left[\frac{\alpha(\frac{1}{2}-\frac{1}{\nu})}{1+\cot(\frac{2\pi}{K})}\cos\Big(\frac{2\pi a}{K}\Big) - \left(1 - \frac{\alpha(\frac{1}{2}-\frac{1}{\nu})}{1+\cot(\frac{2\pi}{K})}\right)\sin\Big(\frac{2\pi a}{K}\Big)\right] \\ &= \frac{2\pi}{K}\sin\Big(\frac{2\pi a}{K}\Big)\left[\alpha\Big(\frac{1}{2}-\frac{1}{\nu}\Big)\cdot\frac{1+\cot(\frac{2\pi a}{K})}{1+\cot(\frac{2\pi}{K})} - 1\right], \end{aligned}$$

which is negative for $a \in \{1,\ldots,K/4\}$ since $\sin(\frac{2\pi a}{K}) > 0$ and $\frac{1+\cot(\frac{2\pi a}{K})}{1+\cot(\frac{2\pi}{K})} \leq 1$ for $K \geq 4$ and $a \in \{1,\ldots,K/4\}$. This completes the proof of (3.98), and hence the lemma.

3.6.4 Proof of Proposition 3.4.2

Recall that $\psi_{\mathsf{c}}^{t+1} = 1 - \mathcal{E}(\tau_{\mathsf{c}}^t)$ where $\mathcal{E}(\tau_{\mathsf{c}}^t)$ is defined in (3.16). We obtain the required lower bound for ψ_{c}^{t+1} by obtaining an upper bound for $\mathcal{E}(\tau_{\mathsf{c}}^t)$. In this section, we drop the dependencies on the column block index c and the iteration index t in our notation for brevity, e.g., we simply denote ν_{c}^t by ν.

We will use the following concentration inequality for the maximum of N i.i.d. standard Gaussian random variables $Z_1, \ldots, Z_N$. For any $\epsilon \in (0,1)$,

$$\mathbb{P}\left(\max_{1\le j\le N} Z_j < \sqrt{2\ln N}(1-\epsilon)\right) \le \exp\left(\frac{-\lambda N^{\epsilon(2-\epsilon)}}{\sqrt{\ln N}}\right), \tag{3.102}$$

where λ is a universal positive constant.

We will prove the required result for the $K \ge 4$ case. The $K = 1$ was proved in [1, Lem. 4.1]. The $K = 2$ case uses similar arguments as the $K \ge 4$ case so we omit the details.

Using the same steps and notations in (3.59)–(3.68), we have that

$$\mathcal{E}(\tau) = \mathbb{E}_{U_1}\mathbb{E}_X\left[\frac{\sum_{k=-K/4+1}^{K/4} 2\cos_k \cdot \sinh Y_k}{X + \sum_{a=-K/4+1}^{K/4} 2\cosh Y_a}\,\middle|\, U_1\right]. \tag{3.103}$$

We will obtain an upper bound for $\mathcal{E}(\tau)$ using the fact that the term inside the expectation in (3.103) is strictly increasing in U_1^R (the real part of $U_1 \sim \mathcal{CN}(0,2)$). (Recall from (3.67) that Y_k is a function of U_1^R for each k.) To see that the term is increasing in U_1^R, we write

$$f := \frac{\sum_{k=-K/4+1}^{K/4} 2\cos_k \cdot \sinh Y_k}{X + \sum_{a=-K/4+1}^{K/4} 2\cosh Y_a} = \frac{f_{\text{num}}}{f_{\text{den}}}, \tag{3.104}$$

and show that $\partial f / \partial U_1^R > 0$. Since

$$\frac{\partial f}{\partial U_1^R} = \Big(\frac{\partial f_{\text{num}}}{\partial U_1^R} \cdot f_{\text{den}} - f_{\text{num}} \cdot \frac{\partial f_{\text{den}}}{\partial U_1^R}\Big) \Big/ \Big(f_{\text{den}}^2\Big), \tag{3.105}$$

we show below that $\frac{\partial f_{\text{num}}}{\partial U_1^R} \cdot f_{\text{dem}} - f_{\text{num}} \cdot \frac{\partial f_{\text{dem}}}{\partial U_1^R} > 0$. Indeed,

$$\begin{aligned}
&\frac{\partial f_{\text{num}}}{\partial U_1^R} \cdot f_{\text{den}} - f_{\text{num}} \cdot \frac{\partial f_{\text{den}}}{\partial U_1^R} \\
&= (\sum_k 2\sqrt{\mu}\cos_k^2 \cosh Y_k)(X + \sum_a 2\cosh Y_a) - (\sum_k 2\cos_k \sinh Y_k)(\sum_a 2\sqrt{\mu}\cos_a \sinh Y_a) \\
&= 4\sqrt{\mu}\left[\frac{X}{2}\cdot(\sum_k \cos_k^2 \cosh Y_k) + (\sum_k \cos_k^2 \cosh Y_k)(\sum_a \cosh Y_a) - (\sum_k \cos_k \sinh Y_k)^2\right] \\
&\overset{\text{(i)}}{\ge} 4\sqrt{\mu}\left[\frac{X}{2}\cdot(\sum_k \cos_k^2 \cosh Y_k) + (\sum_k \cos_k \cosh Y_k)^2 - (\sum_k \cos_k \sinh Y_k)^2\right]
\end{aligned}$$

$$\overset{\text{(ii)}}{>} 4\sqrt{\mu}\cdot\frac{X}{2}\cdot\Big(\sum_k \cos_k^2\cosh Y_k\Big) \;>\; 0,$$

where (i) is obtained using the Cauchy-Schwarz inequality and (ii) from $x^2-y^2=(x+y)(x-y)$ and $\cosh(x)>|\sinh(x)|$ for all x.

Since the term inside the expectation in (3.103) is strictly increasing in U_1^R and upper bounded by 1 (see (3.55)), we have the following upper bound on $\mathcal{E}(\tau)$. Recall that $U_1^R, U_1^I \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and let $\tilde{u}$ be positive scalar variable that will be specified later. Also let $\alpha\in(0,1)$ be a constant that will be specified later.

$$\begin{aligned}
\mathcal{E}(\tau) &\overset{\text{(i)}}{\le} \mathbb{P}(U_1^R>\tilde{u})\cdot 1 \\
&\quad + \mathbb{P}(U_1^R\le\tilde{u})\cdot\mathbb{E}_{U_1^I}\mathbb{E}_X\left[\frac{\sum_{k=-K/4+1}^{K/4} 2\cos_k\cdot\sinh((\mu+\sqrt{\mu}\tilde{u})\cos_k+\sqrt{\mu}U_1^I\sin_k)}{X+\sum_{a=-K/4+1}^{K/4}2\cosh((\mu+\sqrt{\mu}\tilde{u})\cos_a+\sqrt{\mu}U_1^I\sin_a)}\right] \\
&\overset{\text{(ii)}}{\le} Q(\tilde{u})+\mathbb{E}\left[\frac{\sum_{k=-K/4+1}^{K/4}2\cos_k\cdot\sinh(\ldots)}{\sum_{j=2}^{M}\sum_{b=1}^{K}e^{\sqrt{\mu}\left[U_j^R\cos_b+U_j^I\sin_b\right]}+\sum_{a=-K/4+1}^{K/4}2\cosh(\ldots)}\right] \\
&\overset{\text{(iii)}}{\le} Q(\tilde{u})+\mathbb{E}\left[\frac{\sum_{k=-K/4+1}^{K/4}2\cos_k\cdot\sinh(\ldots)}{\big(\max_{2\le j\le M}e^{\sqrt{\nu\ln(KM)}U_j^R}\big)+\sum_{a=-K/4+1}^{K/4}2\cosh(\ldots)}\right] \\
&\le Q(\tilde{u})+\mathbb{P}\Big(\max_{2\le j\le M}U_j^R<\sqrt{2\ln M}(1-\alpha\tilde{\delta})\Big)\cdot 1 \\
&\quad + \mathbb{P}\Big(\max_{2\le j\le M}U_j^R\ge\sqrt{2\ln M}(1-\alpha\tilde{\delta})\Big)\cdot\mathbb{E}_{U_1^I}\left[\frac{\sum_{k=-K/4+1}^{K/4}2\cos_k\cdot\sinh(\ldots)}{M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})}+\sum_{a=-K/4+1}^{K/4}2\cosh(\ldots)}\right] \\
&\overset{\text{(iv)}}{\le} Q(\tilde{u})+e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} \\
&\quad + \mathbb{E}_{U_1^I}\left[\frac{\sum_{k=-K/4+1}^{K/4}2\cos_k\cdot\sinh((\mu+\sqrt{\mu}\tilde{u})\cos_k+\sqrt{\mu}U_1^I\sin_k)}{M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})}+\sum_{a=-K/4+1}^{K/4}2\cosh((\mu+\sqrt{\mu}\tilde{u})\cos_a+\sqrt{\mu}U_1^I\sin_a)}\right], \qquad (3.106)
\end{aligned}$$

where $Q(x)=\int_x^\infty\frac{1}{\sqrt{2\pi}}e^{-z^2/2}\,dz$ is the upper tail probability of the standard Gaussian distribution and the labelled steps are obtained as follows: (i) expanding Y_k according to (3.67) and letting $\tilde{u}>0$ be a parameter we can set later; (ii) using the definition of X in (3.68) (note that the expectation at this step is over $U_2^R,\ldots,U_M^R,U_1^I,\ldots,U_M^I \overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,1)$); (iii) using $\mu=\nu\ln(KM)$ and noting that $\cos_{b=K}=1$, $\sin_{b=K}=0$; and (iv) using (3.102) with λ being a universal positive constant.

We further bound $\mathcal{E}(\tau)$ by noting that the term inside the expectation term on the RHS of (3.106) is bounded by 1.

$$\mathcal{E}(\tau)\;\le\; Q(\tilde{u})+e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}}+\mathbb{P}(U_1^I<-\tilde{u})\cdot 1+\mathbb{P}(U_1^I>\tilde{u})\cdot 1$$

$$
\begin{aligned}
&+ \int_{-\tilde{u}}^{\tilde{u}} \frac{\sum_{k=-K/4+1}^{K/4} 2\cos_k \cdot \sinh((\mu+\sqrt{\mu}\tilde{u})\cos_k + \sqrt{\mu}u\sin_k)}{M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})} + \sum_{a=-K/4+1}^{K/4} 2\cosh((\mu+\sqrt{\mu}\tilde{u})\cos_a + \sqrt{\mu}u\sin_a)} \phi(u)\,du \\
&\overset{\text{(i)}}{\leq} 3Q(\tilde{u}) + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} \\
&\quad + \int_{-\tilde{u}}^{\tilde{u}} \frac{\sum_{k=-K/4+1}^{K/4} 2\sinh((\mu+\sqrt{\mu}\tilde{u})\cos_k + \sqrt{\mu}|u||\sin_k|)}{M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})} + 2\cosh(\mu+\sqrt{\mu}\tilde{u})} \phi(u)\,du \\
&\leq 3Q(\tilde{u}) + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} \\
&\quad + (1-2Q(\tilde{u})) \cdot \left[\frac{\sum_{k=-K/4+1}^{K/4} \exp((\mu+\sqrt{\mu}\tilde{u})\cos_k + \sqrt{\mu}\tilde{u}|\sin_k|)}{M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})} + \exp(\mu+\sqrt{\mu}\tilde{u})} \right] \\
&\overset{\text{(ii)}}{\leq} 3Q(\tilde{u}) + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} + \frac{1+\Delta\cdot\exp(-(\mu+\sqrt{\mu}\tilde{u}))}{1+M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})}\cdot\exp(-(\mu+\sqrt{\mu}\tilde{u}))},
\end{aligned} \tag{3.107}
$$

where $\phi(\cdot)$ denotes the standard Gaussian density and the labelled steps are obtained as follows: (i) noting that $\cos_k \leq 1$, $\cosh(\cdot) > 0$, $\cos_{a=0} = 1$, $\sin_{a=0} = 0$, and that $\sinh(\cdot)$ is an increasing function; and (ii) substituting

$$
\Delta \coloneqq \sum_{k\in\{-K/4+1,\ldots,K/4\}\setminus 0} \exp((\mu+\sqrt{\mu}\tilde{u})\cos_k + \sqrt{\mu}\tilde{u}|\sin_k|). \tag{3.108}
$$

Since the RHS of (3.107) is increasing in Δ, we obtain an upper bound on Δ to further bound $\mathcal{E}(\tau)$.

$$
\begin{aligned}
\Delta &\leq \left(\frac{K}{2}-1\right) \max_{k\in\{-K/4+1,\ldots,K/4\}\setminus 0} \exp((\mu+\sqrt{\mu}\tilde{u})\cos_k + \sqrt{\mu}\tilde{u}|\sin_k|) \\
&\overset{\text{(i)}}{=} \left(\frac{K}{2}-1\right) \max_{k\in\{1,\ldots,K/4\}} \exp((\mu+\sqrt{\mu}\tilde{u})\cos_k + \sqrt{\mu}\tilde{u}\sin_k),
\end{aligned} \tag{3.109}
$$

where step (i) is obtained by noting that taking the maximum over $k \in \{-K/4+1, \ldots, K/4\}\setminus 0$ is the same as taking the maximum over $k \in \{1, \ldots, K/4\}$ since $\cos_k = \cos_{-k}$ and $|\sin_k| = |\sin_{-k}|$.

We now show that for a certain choice of $\tilde{u}$, the maximum in (3.109) is achieved with $k = 1$. For any $\tilde{\alpha} \in (0,1)$, let

$$
\tilde{u} = \frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot(2\pi/K))}\sqrt{\frac{\ln(KM)}{\nu}}, \tag{3.110}
$$

where $\tilde{\kappa} > 0$ is the positive constant defined in (3.23) that lower bounds ν, and $\tilde{\delta} \in (0,1)$ is the constant defined in Proposition 3.4.2. In order to show that $k = 1$ achieves the maximum in (3.109) with this choice of $\tilde{u}$, we show that the derivative of the $\exp(\ldots)$ term with respect to k is negative for all $k \in \{1, \ldots, K/4\}$. Since $\exp(\cdot)$ is an increasing function, we only need to show this negative derivative result for its argument, which we denote by

$$
f_1(k) = (\mu+\sqrt{\mu}\tilde{u})\cos_k + \sqrt{\mu}\tilde{u}\sin_k\,.
$$

Recalling that $\cos_k = \cos(2\pi k/K)$ and $\sin_k = \sin(2\pi k/K)$, its derivative with respect to k is

$$f_1'(k) = \frac{2\pi}{K}\left[\sqrt{\mu}\tilde{u}\cos_k - (\mu+\sqrt{\mu}\tilde{u})\sin_k\right] = \frac{2\pi}{K}\mu\sin_k\left[\frac{\tilde{u}}{\sqrt{\mu}}\cot\left(\frac{2\pi k}{K}\right) - \left(1+\frac{\tilde{u}}{\sqrt{\mu}}\right)\right]$$
$$= \frac{2\pi}{K}\mu\sin_k\left[\frac{\tilde{u}}{\sqrt{\mu}}\left(\cot\left(\frac{2\pi k}{K}\right)-1\right)-1\right] = \frac{2\pi}{K}\mu\sin_k\left[\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2\nu}\cdot\frac{\cot\left(\frac{2\pi k}{K}\right)-1}{\cot\left(\frac{2\pi}{K}\right)+1}-1\right] < 0,$$

where we use (3.110), $\mu = \nu\ln KM$, recalling that $K \geq 4$, and noting that $\sin_k > 0$ for $k \in \{1,\ldots,K/4\}$, and also that $\cot(x)$ is decreasing for $x \in (0, \frac{\pi}{2})$.

Therefore, using the fact that $k = 1$ achieves the maximum in (3.109) when $\tilde{u}$ is chosen according to (3.110), we have an upper bound on Δ which we use in (3.107) to further bound $\mathcal{E}(\tau)$. In the following we use $\cot_1$ to denote $\cot(2\pi/K)$.

$$\begin{aligned}
\mathcal{E}(\tau) &\leq 3Q(\tilde{u}) + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} \\
&\quad + \frac{1+\left(\frac{K}{2}-1\right)\exp((\mu+\sqrt{\mu}\tilde{u})\cos_1 + \sqrt{\mu}\tilde{u}\sin_1)\cdot\exp(-(\mu+\sqrt{\mu}\tilde{u}))}{1+M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})}\cdot\exp(-(\mu+\sqrt{\mu}\tilde{u}))} \\
&\overset{\text{(i)}}{=} 3Q(\tilde{u}) + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} \\
&\quad + \frac{1+\left(\frac{K}{2}-1\right)(KM)^{-\left[\left(\nu+\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot_1)}\right)(1-\cos_1)-\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot_1)}\sin_1\right]}}{1+M^{\sqrt{2\nu}(1-\alpha\tilde{\delta})}\cdot(KM)^{-\left(\nu+\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot_1)}\right)}} \\
&= 3Q(\tilde{u}) + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} + \frac{1+\left(\frac{K}{2}-1\right)(KM)^{-(1-\cos_1)\left(\nu-\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2}\cdot\frac{\frac{\sin_1}{1-\cos_1}-1}{1+\cot_1}\right)}}{1+K^{-\left(\nu+\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot_1)}\right)}\cdot M^{\sqrt{2\nu}\left(1-\left(\alpha+\frac{\tilde{\kappa}\tilde{\alpha}}{2\sqrt{2\nu}(1+\cot_1)}\right)\tilde{\delta}\right)-\nu}} \\
&\overset{\text{(ii)}}{\leq} 3Q(\tilde{u}) + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} + \frac{1+\left(\frac{K}{2}-1\right)(KM)^{-(1-\cos_1)(\nu-\tilde{\kappa}\tilde{\alpha}\tilde{\delta})}}{K^{-\left(\nu+\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot_1)}\right)}\cdot M^{\sqrt{2\nu}\left(1-\left(\alpha+\frac{\tilde{\kappa}\tilde{\alpha}}{2\sqrt{2\nu}(1+\cot_1)}\right)\tilde{\delta}\right)-\nu}} \\
&\overset{\text{(iii)}}{\leq} 3(KM)^{-\frac{(\tilde{\kappa}\tilde{\alpha}\tilde{\delta})^2}{8\nu(1+\cot_1)^2}} + e^{-\lambda(\sqrt{\ln M})^{-1}M^{\alpha\tilde{\delta}(2-\alpha\tilde{\delta})}} \\
&\quad + \frac{K^{\left(\nu+\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot_1)}\right)} + K^{1+\left(\nu+\frac{\tilde{\kappa}\tilde{\alpha}\tilde{\delta}}{2(1+\cot_1)}\right)}\cdot(KM)^{-(1-\cos_1)(\nu-\tilde{\kappa}\tilde{\alpha}\tilde{\delta})}}{M^{\sqrt{2\nu}\left(1-\left(\alpha+\frac{\tilde{\alpha}}{2}\right)\tilde{\delta}\right)-\nu}},
\end{aligned} \tag{3.111}$$

where step (i) is obtained by substituting in (3.110); (ii) is obtained by noting that for $K \geq 4$,

$$\frac{\frac{\sin_1}{1-\cos_1}-1}{1+\cot_1} = \frac{\frac{1-\cos_1^2}{1-\cos_1}-\sin_1}{\cos_1+\sin_1} = \frac{1+\cos_1-\sin_1}{\cos_1+\sin_1} = \frac{1}{\cos_1+\sin_1} + \tan\left(\frac{\pi}{4}-\frac{2\pi}{K}\right) \leq 2, \tag{3.112}$$

and (iii) by using the bound $Q(x) \leq e^{-x^2/2}$ for $x > 0$, and from

$$\frac{\tilde{\kappa}}{2\sqrt{2\nu}(1+\cot_1)} \leq \frac{\tilde{\kappa}}{2\nu} \leq \frac{1}{2}, \tag{3.113}$$

where the inequalities are obtained using $\tilde{\kappa} < \nu < 2$ and $\cot_1 \geq 0$ for $K \geq 4$.

Finally, from the final steps of the proof of the unmodulated ($K = 1$) case in [1, App. A.1],

we know that for $\alpha = 1/64$, $\tilde{\alpha} = 10/32$ and $\nu \in [\tilde{\kappa}, 2 - \tilde{\delta}]$, we have

$$\sqrt{2\nu}\big(1 - \big(\alpha + \frac{\tilde{\alpha}}{2}\big)\tilde{\delta}\big) - \nu \geq \frac{1}{32}\tilde{\delta} + \frac{55}{512}\tilde{\delta}^2. \tag{3.114}$$

By using (3.114) in (3.111) and noting that ν can be upper bounded by a positive constant (see (3.23)), we obtain the required result: for sufficiently large M, any $\tilde{\delta} \in (0, 1)$, and $\nu < 2 - \tilde{\delta}$, we have

$$\mathcal{E}(\tau) \leq M^{-\alpha_{3K}\tilde{\delta}^2},$$

where α_{3K} is a positive constant depending only on K and the bounds of ν in (3.23).

Chapter 4

Many-user Gaussian multiple access

In this chapter, we study the many-user Gaussian multiple access channel (1.6) in the asymptotic regime where the number of users L grows linearly with the code length n. (Throughout this chapter, "asymptotic" refers to the large system limit where L and n both tend to infinity with user density $\mu = L/n$ held constant.) As introduced in Section 1.2.2, we are interested in finding the optimal asymptotic trade-off between the signal-to-noise ratio E_b/N_0 and the user density μ, for a fixed target error rate and number of bits to be transmitted per user (user payload). Here E_b is the energy-per-bit and $N_0 = 2\sigma^2$ is the noise spectral density. Decoding performance is measured via the user error rate (UER):

$$\mathrm{UER} \coloneqq \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\left\{\hat{\boldsymbol{x}}_\ell \neq \boldsymbol{x}_\ell\right\}, \tag{4.1}$$

where $\boldsymbol{x}_\ell$ denotes the message sent by user $\ell \in [L]$, and $\hat{\boldsymbol{x}}_\ell$ is the decoder's estimate of the message. The *per-user probability of error* (PUPE) error criterion used in [32, 33] and defined in (1.10) is the expected value of the UER.

In this chapter, we analyse coding schemes based on random linear models with approximate message passing (AMP) decoding. We derive the exact asymptotic achievable regions of these schemes, and show that the asymptotic achievability of a coding scheme based on *spatially coupled* Gaussian matrices and AMP decoding nearly matches the converse bound for a large range of user densities (Section 4.2). To the best of our knowledge, this is the first efficient coding scheme to do so in this multiple access regime. The spatially coupled scheme can be interpreted as generalised time-sharing: the coupling structure specifies which users are active during each channel use. We also analyse the performance of these coding schemes as the user payload grows large (Section 4.3) and extend our results to *complex* random linear coding schemes for the complex Gaussian multiple access channel (Section 4.4). We show that using small random codebooks multiple times to transmit large user payloads is near-optimal at large user densities, and adding modulation (e.g., K-ary phase shift keying) to the encoding scheme

can increase the size of the asymptotic achievable region in such settings.

4.1 Random linear coding and AMP decoding

We consider coding schemes where the codewords of user $\ell \in [L]$ are constructed as $\boldsymbol{c}_\ell = \boldsymbol{A}_\ell \boldsymbol{x}_\ell$, where $\boldsymbol{A}_\ell \in \mathbb{R}^{n \times B}$ is a random matrix and $\boldsymbol{x}_\ell \in \mathbb{R}^B$ encodes the message of user ℓ. In this coding framework, the Gaussian multiple access channel (MAC) model (1.6) can be written as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \tag{4.2}$$

where the *design matrix* $\boldsymbol{A} \in \mathbb{R}^{n \times LB}$ is the (horizontal) concatenation of matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_L$, and the *message vector* $\boldsymbol{x} \in \mathbb{R}^{LB}$ is the concatenation of vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_L$. We will assume that the squared norm of each column of $\boldsymbol{A}$ equals 1 in expectation.[1]

The L sections of $\boldsymbol{x}$ (which each correspond to a user's message) are drawn i.i.d. from $p_{\boldsymbol{X}_{\text{sec}}}$, which is a probability mass function over a finite set of length B vectors. The per-user payload is therefore equal to the entropy $H(\boldsymbol{X}_{\text{sec}})$, where $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$. The codeword energy constraint is denoted by E, i.e., we require $p_{\boldsymbol{X}_{\text{sec}}}$ to satisfy $\mathbb{E}\|\boldsymbol{X}_{\text{sec}}\|^2 = E < \infty$.

Example 4.1.1 (Random codebooks)**.** Let $p_{\boldsymbol{X}_{\text{sec}}}$ be the distribution over length B vectors that chooses uniformly at random one of its entries to be non-zero, taking the value $\sqrt{E}$. This corresponds to a per-user payload of $\log_2 B$ bits. Then, if the entries of the design matrix $\boldsymbol{A}$ are i.i.d. $\mathcal{N}(0, \frac{1}{n})$, in the coding scheme each user selects one-of-B random codewords of expected energy $\mathbb{E}[\|\boldsymbol{c}_\ell\|^2] = E = E_b \log_2 B$. In the rest of the chapter, we denote the choice of $p_{\boldsymbol{X}_{\text{sec}}}$ used in this example by p_1.

Example 4.1.2 (Random codebooks with binary modulation)**.** Let $p_{\boldsymbol{X}_{\text{sec}}}$ be the distribution over length B vectors that chooses uniformly at random one of its B entries to be non-zero, taking values in $\{\pm\sqrt{E}\}$ with equal probability. This corresponds to a per-user payload of $1 + \log_2 B$ bits. Then, if the entries of the design matrix $\boldsymbol{A}$ are i.i.d. $\mathcal{N}(0, \frac{1}{n})$, in the coding scheme each user encodes $\log_2 B$ bits in the selection of one-of-B random codewords, and an additional 1 bit in whether to flip the sign of the codeword. When $B = 1$, this coding scheme corresponds to random code division multiple access (CDMA) with antipodal signalling. In the rest of the chapter, we denote the choice of $p_{\boldsymbol{X}_{\text{sec}}}$ used in this example by p_2.

4.1.1 Spatially coupled coding schemes

Spatially coupled matrices constructed via base matrices are described in detail for the SPARC construction in Section 2.1, we will give an overview of the construction here in the context of

[1] Compared to earlier chapters, this chapter uses $\boldsymbol{x}$ instead of $\boldsymbol{\beta}$ to denote the message vector, and B instead of M to denote the size of each section of the message vector. This is done so that our notation is compatible (and not easily confused) with existing notation in the many-user Gaussian MAC literature.

coding for Gaussian MACs. A spatially coupled (Gaussian) design matrix $\boldsymbol{A} \in \mathbb{R}^{n \times LB}$ is divided into R-by-C equally size *blocks*. The entries within each block are i.i.d. Gaussian with zero mean and variance specified by the corresponding entry of a base matrix $\boldsymbol{W} \in \mathbb{R}_+^{\mathsf{R} \times \mathsf{C}}$. The design matrix $\boldsymbol{A}$ is constructed by replacing each entry of the base matrix W_{rc}, by an $(n/\mathsf{R}) \times (LB/\mathsf{C})$ matrix with entries drawn i.i.d. from $\mathcal{N}(0, W_{\mathsf{rc}}/(n/\mathsf{R}))$. See Fig. 2.1 for an example. Hence, the design matrix $\boldsymbol{A}$ has independent Gaussian entries

$$A_{ij} \sim \mathcal{N}\bigg(0, \frac{1}{n/\mathsf{R}} W_{\mathsf{r}(i)\mathsf{c}(j)}\bigg), \quad \text{for } i \in [n],\ j \in [LB]. \tag{4.3}$$

The operators $\mathsf{r}(\cdot) : [n] \to [\mathsf{R}]$ and $\mathsf{c}(\cdot) : [LB] \to [\mathsf{C}]$ in (4.3) map a particular row or column index to its corresponding *row block* or *column block* index. We require C to divide L, resulting in $L/\mathsf{C} \geq 1$ sections per column block.

The entries of the base matrix $\boldsymbol{W}$ must satisfy $\sum_{\mathsf{r}=1}^{\mathsf{R}} W_{\mathsf{rc}} = 1$ for $\mathsf{c} \in [\mathsf{C}]$ to ensure that the columns of the design matrix $\boldsymbol{A}$ have unit norm in expectation. The trivial base matrix with $\mathsf{R} = \mathsf{C} = 1$ (single entry equal to 1) corresponds to the design matrix with i.i.d. $\mathcal{N}(0, \frac{1}{n})$ entries. In this chapter we will consider a class of base matrices called (ω, Λ, ρ) base matrices [1,2]. (Note that the definition differs slightly from that in Definition 2.1.1 for spatially coupled SPARCs due to the different constraint on the base matrix entries.)

Definition 4.1.1. An (ω, Λ, ρ) base matrix $\boldsymbol{W}$ is described by three parameters: coupling width $\omega \geq 1$, coupling length $\Lambda \geq 2\omega - 1$, and $\rho \in [0, 1)$ which specifies the fraction of "energy" allocated to the uncoupled entries in each column. The matrix has $\mathsf{R} = \Lambda + \omega - 1$ rows and $\mathsf{C} = \Lambda$ columns, with each column having ω identical non-zero entries in the band-diagonal region. The (r, c)-th entry of the base matrix, for $\mathsf{r} \in [\mathsf{R}], \mathsf{c} \in [\mathsf{C}]$, is given by

$$W_{\mathsf{rc}} = \begin{cases} \frac{1-\rho}{\omega} & \text{if } \mathsf{c} \leq \mathsf{r} \leq \mathsf{c} + \omega - 1, \\ \frac{\rho}{\Lambda - 1} & \text{otherwise.} \end{cases} \tag{4.4}$$

When $\rho = 0$, the base matrix has non-zero entries only in the band-diagonal region. For example, the base matrix in Fig. 2.1 has parameters ($\omega = 3$, $\Lambda = 7$, $\rho = 0$).

Each entry of the base matrix corresponds to an $(n/\mathsf{R}) \times (LB/\mathsf{C})$ block of the design matrix $\boldsymbol{A}$, and each block can be viewed as an (uncoupled) i.i.d. Gaussian design matrix with L/C sections, code length n/R, and user density

$$\mu_{\text{inner}} = \frac{L/\mathsf{C}}{n/\mathsf{R}} = \frac{\mathsf{R}}{\mathsf{C}}\,\mu = \bigg(1 + \frac{\omega - 1}{\Lambda}\bigg)\,\mu. \tag{4.5}$$

Since $\omega > 1$ in spatially coupled systems, we have $\mu < \mu_{\text{inner}}$. This difference is often referred to as a "rate loss" in the literature of spatially coupled error correcting codes [2, 14, 15, 131], and becomes negligible when Λ is much larger than ω.

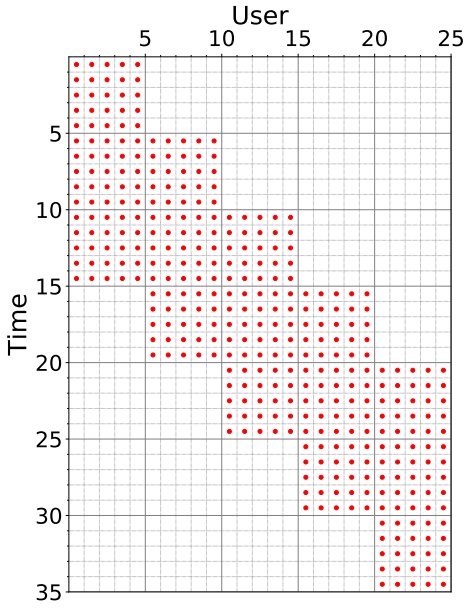


Figure 4.1: An example of how 25 users communicate over 35 uses of the channel using the multiple access scheme based on a spatially coupled design matrix constructed using an ($\omega = 3$, $\Lambda = 5$, $\rho = 0$) base matrix. A red dot in the 2D grid represents a certain user being active during a certain time instant and empty squares represent silence.

The spatially coupled coding scheme can be viewed as *block-wise time-division with overlap*. Consider a scenario with $L = 25$ users, $n = 35$ channel uses, and a spatially coupled design matrix constructed using an ($\omega = 3$, $\Lambda = 5$, $\rho = 0$) base matrix $\boldsymbol{W} \in \mathbb{R}_{+}^{7\times 5}$. Each block of the design matrix corresponds to 5 channel uses and 5 users. Fig. 4.1 shows how users communicate using this multiple access scheme, assuming the code dimension is time and that each channel use corresponds to one time instant. A red dot in the 2D grid represents a certain user being active (transmitting) during a certain time instant, and empty squares represent silence (no transmission). For example, users in the first column block (users 1 to 5) will transmit during time instants 1 to 15 (corresponding to the first $\omega = 3$ row blocks) but silent afterwards; users 6 to 10 will transmit during time instants 6 to 20 but silent otherwise, and so on.

Users within the same column block transmit simultaneously over $\omega = 3$ row blocks of time (15 time instants), and users in neighbouring column blocks overlap in $\omega - 1$ row blocks of time. At each time instant, $\omega = 3$ column blocks of users (15 users) simultaneously transmit (less users at the initial and end time instances), but the set of active users gradually shifts over time. Thus, this multiple access scheme can be seen as a (block-wise) time-division with overlap scheme. When $\omega = 1$, there is no time overlap (no coupling) between neighbouring blocks of users and each block of users communicates using an i.i.d. Gaussian matrix. When Λ is large with respect to ω, users are silent for most of the transmission period (n channel uses). This facilitates low-complexity encoding and decoding (e.g., the sliding window AMP decoder introduced in Section 2.6).

4.1.2 AMP decoding and state evolution

We consider an efficient AMP decoder that aims to reconstruct the message vector $\boldsymbol{x}$ from the channel output $\boldsymbol{y}$. The design matrix $\boldsymbol{A}$, the base matrix $\boldsymbol{W}$, the distribution $p_{\boldsymbol{X}_{\text{sec}}}$, and the channel noise variance σ^2 are known to the decoder. The AMP decoder we now describe is almost identical to the AMP decoder for spatially coupled SPARCs described in Section 2.2, albeit with a slight change in the scaling of certain variables related to the energy E, and considering the user density μ instead of the code rate R.

The AMP decoder iteratively generates message vector estimates $\boldsymbol{x}^t \in \mathbb{R}^{LB}$ for iterations $t = 1, 2, \dots$ as follows. Initialise $\boldsymbol{x}^0$ to the all-zero vector, and for $t \geq 0$, iteratively compute:

$$\begin{aligned} \boldsymbol{z}^t &= \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^t + \widetilde{\boldsymbol{v}}^t \odot \boldsymbol{z}^{t-1}, \\ \boldsymbol{x}^{t+1} &= \eta^t\big(\boldsymbol{x}^t + (\widetilde{\boldsymbol{Q}}^t \odot \boldsymbol{A})^* \boldsymbol{z}^t\big). \end{aligned} \tag{4.6}$$

Here $\odot$ is the Hadamard (entry-wise) product and quantities with negative iteration indices are set to zero. The vector $\widetilde{\boldsymbol{v}}^t \in \mathbb{R}^n$ and the matrix $\widetilde{\boldsymbol{Q}} \in \mathbb{R}^{n \times LB}$ will be described in terms of the following state evolution parameters.

State evolution The performance of the AMP in the large system limit ($L, n \to \infty$ with $\mu = L/n$ held constant) is succinctly captured by a deterministic recursion called *state evolution* (SE). State evolution iteratively defines vectors $\boldsymbol{\gamma}^t, \boldsymbol{\phi}^t \in \mathbb{R}^{\mathsf{R}}$ and $\boldsymbol{\tau}^t, \boldsymbol{\psi}^t \in \mathbb{R}^{\mathsf{C}}$ as follows. Initialise $\psi_{\mathsf{c}}^0 = E$ for $\mathsf{c} \in [\mathsf{C}]$, and for $t \geq 0$, iteratively compute:

$$\gamma_{\mathsf{r}}^t = \sum_{\mathsf{c}=1}^{\mathsf{C}} W_{\mathsf{rc}} \psi_{\mathsf{c}}^t, \qquad \phi_{\mathsf{r}}^t = \sigma^2 + \mu_{\text{inner}} \gamma_{\mathsf{r}}^t, \qquad \mathsf{r} \in [\mathsf{R}], \tag{4.7}$$

$$\tau_{\mathsf{c}}^t = \left[\sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\phi_{\mathsf{r}}^t}\right]^{-1}, \qquad \psi_{\mathsf{c}}^{t+1} = \mathsf{mmse}(1/\tau_{\mathsf{c}}^t), \qquad \mathsf{c} \in [\mathsf{C}], \tag{4.8}$$

where $\mu_{\text{inner}} = \frac{\mathsf{R}}{\mathsf{C}}\,\mu$ from (4.5), and

$$\mathsf{mmse}(1/\tau) = \mathbb{E}\left\| \boldsymbol{X}_{\text{sec}} - \mathbb{E}\Big[\boldsymbol{X}_{\text{sec}} \mid \boldsymbol{X}_{\text{sec}} + \sqrt{\tau}\boldsymbol{Z}\Big] \right\|^2 \tag{4.9}$$

$$= \begin{cases} E\left[1 - \mathbb{E}\left[\dfrac{e^{\sqrt{\frac{E}{\tau}} Z_1}}{e^{\sqrt{\frac{E}{\tau}} Z_1} + e^{-E/\tau} \sum_{j=2}^{B} e^{\sqrt{\frac{E}{\tau}} Z_j}}\right]\right] & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_1, \\ E\left[1 - \mathbb{E}\left[\dfrac{\sinh(\frac{E}{\tau} + \sqrt{\frac{E}{\tau}} Z_1)}{\cosh(\frac{E}{\tau} + \sqrt{\frac{E}{\tau}} Z_1) + \sum_{j=2}^{B} \cosh(\sqrt{\frac{E}{\tau}} Z_j)}\right]\right] & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_2, \end{cases} \tag{4.10}$$

where $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$ and $\boldsymbol{Z} = [Z_1, \dots, Z_B]$ is a standard Gaussian vector independent of $\boldsymbol{X}_{\text{sec}}$. Recall that p_1 and p_2 are described in Examples 4.1.1 and 4.1.2.

The vector $\widetilde{\boldsymbol{v}}^t \in \mathbb{R}^n$ and the matrix $\widetilde{\boldsymbol{Q}}^t \in \mathbb{R}^{n \times LB}$ in (4.6) both have a block-wise structure

and are defined using state evolution parameters as follows. For $i \in [n]$ and $j \in [LB]$,

$$\widetilde{v}_i^t = \frac{\mu_{\text{inner}}\, \gamma_{\mathsf{r}(i)}^t}{\phi_{\mathsf{r}(i)}^{t-1}}, \qquad \widetilde{Q}_{ij}^t = \frac{\tau_{\mathsf{c}(j)}^t}{\phi_{\mathsf{r}(i)}^t}, \tag{4.11}$$

where we recall that $\mathsf{r}(i)$ and $\mathsf{c}(j)$ denote the row and column block index of the i-th row entry and j-th column entry, respectively. The vector $\widetilde{\boldsymbol{v}}^0$ is defined to be all-zeros.

Denoising function In each iteration, the AMP decoder (4.6) produces an effective observation $\boldsymbol{s}^t = \boldsymbol{x}^t + (\widetilde{\boldsymbol{Q}}^t \odot \boldsymbol{A})^* \boldsymbol{z}^t$, which has the following approximate representation: for an index j in column block c of the message vector $\boldsymbol{x}$, we have $s_j^t \approx x_j + \sqrt{\tau_{\mathsf{c}}^t} Z_j$, where $\{Z_j\}$ i.i.d. $\sim \mathcal{N}(0,1)$. The estimate $\boldsymbol{x}^{t+1}$ in (4.6) is then the minimum mean square error (MMSE) estimate of $\boldsymbol{x}$ given $\boldsymbol{s}^t$, computed using the assumed distribution. This leads to the following definition of the denoising function $\eta^t = (\eta_1^t, \ldots, \eta_{LB}^t)$ in (4.6): for index j in section $\ell \in [L]$, which we denote by $j \in \text{sec}(\ell)$,

$$\eta_j^t(\boldsymbol{s}) = \mathbb{E}\left[(\boldsymbol{X}_{\text{sec}})_j \mid \boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}(j)}^t}\, \boldsymbol{Z} = \boldsymbol{s}_\ell \right] \tag{4.12}$$

$$= \begin{cases} \sqrt{E} \cdot \dfrac{\exp\left(s_j \sqrt{E}/\tau_{\mathsf{c}(j)}^t\right)}{\sum_{i \in \text{sec}(\ell)} \exp\left(s_i \sqrt{E}/\tau_{\mathsf{c}(j)}^t\right)} & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_1, \\ \sqrt{E} \cdot \dfrac{\sinh\left(s_j \sqrt{E}/\tau_{\mathsf{c}(j)}^t\right)}{\sum_{i \in \text{sec}(\ell)} \cosh\left(s_i \sqrt{E}/\tau_{\mathsf{c}(j)}^t\right)} & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_2, \end{cases} \tag{4.13}$$

where we recall that the ℓ-th section of a vector $\boldsymbol{s} \in \mathbb{R}^{LB}$ is denoted by $\boldsymbol{s}_\ell \in \mathbb{R}^B$.

Hard decision estimate In addition to $\boldsymbol{x}^{t+1}$, the decoder can also produce a *hard-decision* maximum a posteriori (MAP) estimate from $\boldsymbol{s}^t$, which we denote by $\hat{\boldsymbol{x}}^{t+1}$. For section ℓ in column block $\mathsf{c} \in [\mathsf{C}]$, the ℓ-th section of this hard-decision estimate is given by

$$\hat{\boldsymbol{x}}_\ell^{t+1} = \arg\max_{\boldsymbol{x}' \in \mathcal{S}} \mathbb{P}\Big(\boldsymbol{X}_{\text{sec}} = \boldsymbol{x}' \mid \boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\, \boldsymbol{Z} = \boldsymbol{s}_\ell^t \Big), \tag{4.14}$$

where $\mathcal{S}$ is the support of $p_{\boldsymbol{X}_{\text{sec}}}$. When $\boldsymbol{X}_{\text{sec}} \sim p_1$, index $j \in \text{sec}(\ell)$ of this hard-decision estimate is given by

$$\hat{x}_j^{t+1} = \begin{cases} \sqrt{E} & \text{if } s_j^t > s_i^t \text{ for all } i \in \text{sec}(\ell) \backslash j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

When $\boldsymbol{X}_{\text{sec}} \sim p_2$, index $j \in \text{sec}(\ell)$ of this hard-decision estimate is given by

$$\hat{x}_j^{t+1} = \begin{cases} \text{sign}(s_j) \cdot \sqrt{E} & \text{if } |s_j^t| > |s_i^t| \text{ for all } i \in \text{sec}(\ell) \backslash j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

AMP and SE for i.i.d. Gaussian $\boldsymbol{A}$ For the special case where the entries of the design matrix $\boldsymbol{A}$ are i.i.d. $\mathcal{N}(0, \frac{1}{n})$, the AMP decoder (4.6) and the state evolution (4.7)–(4.8) can be simplified. The AMP decoder initialises the message vector estimate $\boldsymbol{x}^0$ to the all-zero vector, and for $t \geq 0$, iteratively computes:

$$\begin{aligned} \boldsymbol{z}^t &= \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^t + \frac{\mu\psi^t}{\tau^{t-1}}\, \boldsymbol{z}^{t-1}, \\ \boldsymbol{x}^{t+1} &= \eta^t\big(\boldsymbol{x}^t + \boldsymbol{A}^*\boldsymbol{z}^t\big), \end{aligned} \tag{4.17}$$

where quantities with negative iteration indices are set to zero. The scalars τ^t and ψ^t are given by the state evolution. The state evolution initialises $\psi^0 = E$, and for $t \geq 0$, iteratively computes:

$$\begin{aligned} \tau^t &= \sigma^2 + \mu\psi^t, \\ \psi^{t+1} &= \mathsf{mmse}(1/\tau^t), \end{aligned} \tag{4.18}$$

where the mmse function is defined in (4.9). Furthermore, when the design matrix has i.i.d. Gaussian entries, the denoising function η^t in (4.12) and the hard-decision estimate $\hat{\boldsymbol{x}}^{t+1}$ in (4.14) are defined using the state evolution parameter τ^t as the effective noise variance.

4.2 Asymptotic UER achieved by AMP decoding

We now characterise the asymptotic user error rate (4.1) achieved by coding schemes based on i.i.d. and spatially coupled Gaussian design matrices with AMP decoding. These results are stated in terms of a *potential function*.

4.2.1 Potential function

Consider the single-section Gaussian channel with noise variance τ:

$$\boldsymbol{S}_\tau = \boldsymbol{X}_{\text{sec}} + \sqrt{\tau}\boldsymbol{Z}, \tag{4.19}$$

where $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$ and $\boldsymbol{Z} \in \mathbb{R}^B$ is a standard Gaussian vector independent of $\boldsymbol{X}_{\text{sec}}$. The potential function for the random linear system (4.2) with user density $\mu = L/n$ and channel

noise variance σ^2 is defined as

$$\mathcal{F}(\mu, \sigma^2, \psi) = I(\boldsymbol{X}_{\text{sec}}; \boldsymbol{S}_\tau) + \frac{1}{2\mu}\left[\ln\left(\frac{\tau}{\sigma^2}\right) - \frac{\mu\psi}{\tau}\right], \tag{4.20}$$

where $\psi \in [0, E]$, $\tau = \sigma^2 + \mu\psi$, and the mutual information $I(\boldsymbol{X}_{\text{sec}}; \boldsymbol{S}_\tau)$ is computed using the channel (4.19). If $\boldsymbol{Z} = [Z_1, \ldots, Z_B]$, then for the specific choices of $p_{\boldsymbol{X}_{\text{sec}}}$ given in Examples 4.1.1 and 4.1.2, we have

$$I(\boldsymbol{X}_{\text{sec}}; \boldsymbol{S}_\tau) = \begin{cases} \frac{E}{\tau} + \ln B - \mathbb{E}\ln\left[\exp\left(\frac{E}{\tau} + \sqrt{\frac{E}{\tau}}Z_1\right) + \sum_{j=2}^{B}\exp\left(\sqrt{\frac{E}{\tau}}Z_j\right)\right] & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_1, \\ \frac{E}{\tau} + \ln B - \mathbb{E}\ln\left[\cosh\left(\frac{E}{\tau} + \sqrt{\frac{E}{\tau}}Z_1\right) + \sum_{j=2}^{B}\cosh\left(\sqrt{\frac{E}{\tau}}Z_j\right)\right] & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_2. \end{cases} \tag{4.21}$$

Define the set of potential function minimisers (w.r.t. ψ) as:

$$\mathcal{M}(\mu, \sigma^2) = \left\{ \underset{\psi \in [0,E]}{\arg\min}\ \mathcal{F}(\mu, \sigma^2, \psi) \right\}. \tag{4.22}$$

Consider decoding $\boldsymbol{X}_{\text{sec}}$ in the Gaussian channel in (4.19). The MMSE decoder

$$\hat{\boldsymbol{x}}_{\text{sec}}^{\text{MMSE}}(\boldsymbol{S}_\tau) = \mathbb{E}[\boldsymbol{X}_{\text{sec}}|\boldsymbol{S}_\tau], \tag{4.23}$$

achieves the MMSE given by (4.9). The MAP decoder

$$\hat{\boldsymbol{x}}_{\text{sec}}^{\text{MAP}}(\boldsymbol{S}_\tau) = \underset{\boldsymbol{x}'}{\arg\max}\, \mathbb{P}(\boldsymbol{X}_{\text{sec}} = \boldsymbol{x}'|\boldsymbol{S}_\tau), \tag{4.24}$$

achieves the minimum probability of error, given by

$$P_e(\tau) = \mathbb{P}\big(\hat{\boldsymbol{x}}_{\text{sec}}^{\text{MAP}}(\boldsymbol{S}_\tau) \neq \boldsymbol{X}_{\text{sec}}\big) \tag{4.25}$$

$$= \begin{cases} 1 - \mathbb{E}\left[\Phi\left(\sqrt{\frac{E}{\tau}} + Z\right)^{B-1}\right] & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_1, \\ 1 - \displaystyle\int_{-\sqrt{\frac{E}{\tau}}}^{\infty}\left[1 - 2Q\left(\sqrt{\frac{E}{\tau}} + z\right)\right]^{B-1}\phi(z)\,dz & \text{if } \boldsymbol{X}_{\text{sec}} \sim p_2, \end{cases} \tag{4.26}$$

where $Z \sim \mathcal{N}(0,1)$. The functions $\phi(\cdot)$, $\Phi(\cdot)$ and $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-z^2/2}\,dz$ are the probability density function, cumulative distribution function and upper tail probability of the standard Gaussian distribution, respectively.

4.2.2 I.I.D. Gaussian matrices

Theorem 4 (I.I.D. Gaussian matrices with AMP decoding)**.** *Consider the linear model* (4.2)*, with the entries of the design matrix* $\boldsymbol{A}$ *i.i.d.* $\sim \mathcal{N}(0, \frac{1}{n})$ *and the* L *sections of the message vector* $\boldsymbol{x}$ *i.i.d.* $\sim p_{\boldsymbol{X}_{sec}}$*. Let* $\hat{\boldsymbol{x}}^t$ *be the AMP hard-decision estimate of* $\boldsymbol{x}$ *after iteration* t *(defined in* (4.14)*), and recall that* τ^t, ψ^t *are outputs of the state evolution* (4.18)*.*

1) The sequences $\{\tau^t\}_{t\geq 0}$ *and* $\{\psi^t\}_{t\geq 0}$ *are non-increasing and converge to fixed points* τ^{FP}*,* ψ^{FP}*, where*

$$\tau^{FP} \coloneqq \sigma^2 + \mu\psi^{FP}, \tag{4.27}$$

$$\psi^{FP} \coloneqq \max\left\{\psi : \psi = \mathit{mmse}\left(\frac{1}{\sigma^2 + \mu\psi}\right)\right\} = \max\left\{\psi : \frac{\partial \mathcal{F}(\mu, \sigma^2, \psi)}{\partial \psi} = 0\right\}. \tag{4.28}$$

The potential function $\mathcal{F}(\mu, \sigma^2, \psi)$ *is defined in* (4.20)*.*

2) Fix $\delta > 0$*, and let* T *denote the first iteration for which* $\tau^t \leq \tau^{FP} + \delta$*. Then the user error rate of the AMP decoder after* $T + 1$ *iterations satisfies*

$$\lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{T+1} \neq \boldsymbol{x}_\ell\} \overset{a.s.}{=} P_e(\tau^T) \leq P_e(\tau^{FP} + \delta), \tag{4.29}$$

where the limit is taken with $\frac{L}{n} = \mu$ *held constant and* $P_e(\cdot)$ *is defined in* (4.25)*.*

Proof. 1) We first prove that the sequence $\{\psi^t\}_{t\geq 0}$ is non-increasing and converges to the fixed point ψ^{FP} defined in (4.28). Then the result that $\{\tau^t\}_{t\geq 0}$ is non-increasing and converges to $\tau^{\mathrm{FP}} = \sigma^2 + \mu\psi^{\mathrm{FP}}$ immediately follows since $\tau^t = \sigma^2 + \mu\psi^t$ (an increasing function of ψ^t).

Let us consider the state evolution (4.18) as a single recursion:

$$\psi^{t+1} = \mathsf{mmse}((\sigma^2 + \mu\psi^t)^{-1}). \tag{4.30}$$

Starting from $\psi^0 = \mathbb{E}\|\boldsymbol{X}_{\mathrm{sec}}\|^2 = E$, we have that

$$\psi^1 = \mathsf{mmse}((\sigma^2 + \mu E)^{-1}) \leq E = \psi^0, \tag{4.31}$$

where the inequality is because the trivial all zero estimate of a random section $\boldsymbol{X}_{\mathrm{sec}}$ achieves an expected squared error of E. The mmse function defined in (4.9) is a non-increasing function of its argument snr — this has been rigorously shown in [140, 163]. Since its argument $\mathsf{snr} = (\sigma^2 + \mu\psi^t)^{-1}$ is decreasing in ψ^t, the mmse function is non-decreasing in ψ^t. Therefore, if $\psi^t \leq \psi^{t-1}$, then

$$\psi^{t+1} = \mathsf{mmse}((\sigma^2 + \mu\psi^t)^{-1}) \leq \mathsf{mmse}((\sigma^2 + \mu\psi^{t-1})^{-1}) = \psi^t,$$

which together with (4.31) shows that the sequence $\{\psi^t\}_{t\geq 0}$ is non-increasing. Moreover, if

$\psi^t \geq \psi^{\mathrm{FP}}$, then $\psi^{t+1} \geq \psi^{\mathrm{FP}}$. Indeed, for any $\psi^t \geq \psi^{\mathrm{FP}}$,

$$\psi^{t+1} = \mathsf{mmse}((\sigma^2 + \mu\psi^t)^{-1}) \geq \mathsf{mmse}((\sigma^2 + \mu\psi^{\mathrm{FP}})^{-1}) = \psi^{\mathrm{FP}}.$$

Since $\{\psi^t\}_{t\geq 0}$ is a non-increasing sequence bounded below by ψ^{FP} (noting that $\psi^0 \geq \psi^{\mathrm{FP}}$), we conclude that it converges to ψ^{FP}.

To show that the fixed points of the state evolution correspond to the stationary points of the potential function $\mathcal{F}(\mu, \sigma^2, \psi)$ defined in (4.20), we compute the derivative:

$$\begin{aligned}
\frac{\partial \mathcal{F}(\mu,\sigma^2,\psi)}{\partial \psi} &= \frac{\partial \tau^{-1}}{\partial \psi}\frac{\partial}{\partial \tau^{-1}} I(\boldsymbol{X}_{\mathrm{sec}}; \boldsymbol{X}_{\mathrm{sec}} + \sqrt{\tau}\boldsymbol{Z}) + \frac{\partial}{\partial \psi}\frac{1}{2\mu}\left[\ln\left(1+\frac{\mu\psi}{\sigma^2}\right) - \frac{\mu\psi}{\sigma^2+\mu\psi}\right] \\
&= \frac{-\mu}{(\sigma^2+\mu\psi)^2}\,\frac{1}{2}\,\mathsf{mmse}(1/\tau) + \frac{\mu\psi}{2(\sigma^2+\mu\psi)^2} \\
&= \frac{\mu}{2(\sigma^2+\mu\psi)^2}\left[\psi - \mathsf{mmse}\left(\frac{1}{\sigma^2+\mu\psi}\right)\right],
\end{aligned}$$

where the equalities are obtained using $\tau = \sigma^2 + \mu\psi$ and the vector I-MMSE relationship [85, Thm. 2]. Therefore, since $\sigma^2 > 0$ and $\mu > 0$, we have that $\partial\mathcal{F}(\mu,\sigma^2,\psi)/\partial\psi = 0$ corresponds to $\psi = \mathsf{mmse}((\sigma^2+\mu\psi)^{-1})$, which is the fixed point of the iteration (4.30).

2) We now prove (4.29). For $\ell \in [L]$, we denote by $\boldsymbol{a}_\ell \in \mathbb{R}^B$ the ℓ-th section of a vector $\boldsymbol{a} \in \mathbb{R}^{LB}$. Consider the input to the AMP hard-decision step in iteration $t+1$, which we denote by $\boldsymbol{s}^t = \boldsymbol{x}^t + \boldsymbol{A}^*\boldsymbol{z}^t \in \mathbb{R}^{LB}$ (see (4.14), (4.17)). The MAP estimator $\hat{\boldsymbol{x}}_\ell^{t+1} = \hat{\boldsymbol{x}}_\ell^{t+1}(\boldsymbol{s}_\ell^t)$ in (4.14) partitions the space $\mathbb{R}^B$ into decision regions. For each $\boldsymbol{x}_\ell$ in the support of $p_{\boldsymbol{X}_{\mathrm{sec}}}$, the decision region is

$$\mathcal{D}(\boldsymbol{x}_\ell) := \left\{\boldsymbol{s}_\ell^t : \hat{\boldsymbol{x}}_\ell^{t+1}(\boldsymbol{s}_\ell^t) = \boldsymbol{x}_\ell\right\}. \tag{4.32}$$

Note that $\mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{t+1}(\boldsymbol{s}_\ell^t) = \boldsymbol{x}_\ell\} = \mathbb{1}\{\boldsymbol{s}_\ell^t \in \mathcal{D}(\boldsymbol{x}_\ell)\}$.

The distance between a vector $\boldsymbol{v} \in \mathbb{R}^B$ and a set $\mathcal{B} \subset \mathbb{R}^B$ is denoted by $d(\boldsymbol{v}, \mathcal{B}) := \inf\{\|\boldsymbol{v} - \boldsymbol{u}\|_2 : \boldsymbol{u} \in \mathcal{B}\}$. For any $\epsilon > 0$, define the $\psi_{\epsilon,+}, \psi_{\epsilon,-} : \mathbb{R}^B \times \mathbb{R}^B \to \mathbb{R}$ as follows:

$$\psi_{\epsilon,+}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t) = \begin{cases} 1, & \boldsymbol{s}_\ell^t \in \mathcal{D}(\boldsymbol{x}_\ell), \\ 0, & d(\boldsymbol{s}_\ell^t, \mathcal{D}(\boldsymbol{x}_\ell)) > \epsilon, \\ 1 - d(\boldsymbol{s}_\ell^t, \mathcal{D}(\boldsymbol{x}_\ell))/\epsilon, & \text{otherwise}, \end{cases} \tag{4.33}$$

$$\psi_{\epsilon,-}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t) = \begin{cases} 1, & d(\boldsymbol{s}_\ell^t, \mathcal{D}(\boldsymbol{x}_\ell)^c) > \epsilon, \\ 0, & \boldsymbol{s}_\ell^t \in \mathcal{D}(\boldsymbol{x}_\ell)^c, \\ d(\boldsymbol{s}_\ell^t, \mathcal{D}(\boldsymbol{x}_\ell)^c)/\epsilon, & \text{otherwise}. \end{cases} \tag{4.34}$$

We note that $\psi_{\epsilon,+}, \psi_{\epsilon,-}$ are Lipschitz-continuous (with Lipschitz constant $1/\epsilon$), and

$$\psi_{\epsilon,-}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t) \leq \mathbb{1}\{\boldsymbol{s}_\ell^t \in \mathcal{D}(\boldsymbol{x}_\ell)\} \leq \psi_{\epsilon,+}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t),$$

and thus

$$\frac{1}{L}\sum_{\ell=1}^{L} \psi_{\epsilon,-}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t) \leq \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{t+1}(\boldsymbol{s}_\ell^t) = \boldsymbol{x}_\ell\} \leq \frac{1}{L}\sum_{\ell=1}^{L} \psi_{\epsilon,+}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t). \tag{4.35}$$

The results in [76] and [1,57] imply that for any pseudo-Lipschitz function $\psi : \mathbb{R}^B \times \mathbb{R}^B \to \mathbb{R}$, the following holds almost surely:

$$\lim_{L\to\infty} \frac{1}{L}\sum_{\ell=1}^{L} \psi(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t) = \mathbb{E}\{\psi(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau^t})\}, \tag{4.36}$$

where $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$ and $\boldsymbol{S}_{\tau^t}$ is given by (4.19). This result was proved in [76] for the case $B = 1$, and extended in [1, 57] to the setting of sparse regression codes where the specific distribution $p_{\boldsymbol{X}_{\text{sec}}}$ given in Example 4.1.1 (corresponding to random codebooks) is used. The proof for more general discrete distributions is essentially the same. In (4.36) and in the equations below, $L/n = \mu$ as $L \to \infty$.

Applying (4.36) to the Lipschitz continuous functions $\psi_{\epsilon,+}$ and $\psi_{\epsilon,-}$, we obtain:

$$\begin{aligned}
\lim_{L\to\infty} \frac{1}{L}\sum_{\ell}^{L} \psi_{\epsilon,+}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t) &= \mathbb{E}\{\psi_{\epsilon,+}(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau^t})\} \ \text{ a.s.} \\
\lim_{L\to\infty} \frac{1}{L}\sum_{\ell}^{L} \psi_{\epsilon,-}(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t) &= \mathbb{E}\{\psi_{\epsilon,-}(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau^t})\} \ \text{ a.s.}
\end{aligned} \tag{4.37}$$

Since $\epsilon > 0$ is arbitrary, from (4.35) and (4.37), we almost surely have

$$\begin{aligned}
\lim_{\epsilon\to 0} \mathbb{E}\{\psi_{\epsilon,-}(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau^t})\} &\leq \liminf_{L\to\infty} \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{t+1}(\boldsymbol{s}_\ell^t) = \boldsymbol{x}_\ell\} \\
&\leq \limsup_{L\to\infty} \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{t+1}(\boldsymbol{s}_\ell^t) = \boldsymbol{x}_\ell\} \leq \lim_{\epsilon\to 0} \mathbb{E}\{\psi_{\epsilon,+}(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau^t})\}.
\end{aligned} \tag{4.38}$$

By the monotone convergence theorem, we have

$$\begin{aligned}
\lim_{\epsilon\to 0} \mathbb{E}\{\psi_{\epsilon,-}(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau^t})\} &= \mathbb{P}(\boldsymbol{S}_{\tau^t} \in \mathcal{D}(\boldsymbol{X}_{\text{sec}})) = 1 - P_e(\tau^t), \\
\lim_{\epsilon\to 0} \mathbb{E}\{\psi_{\epsilon,+}(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau^t})\} &= \mathbb{P}(\boldsymbol{S}_{\tau^t} \in \mathcal{D}(\boldsymbol{X}_{\text{sec}})) = 1 - P_e(\tau^t).
\end{aligned} \tag{4.39}$$

This completes the proof that

$$\lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{t+1}(\boldsymbol{s}_\ell^t) \neq \boldsymbol{x}_\ell\} = P_e(\tau^t). \tag{4.40}$$

□

Remark 4.2.1 (Related work)**.** In [36, Sec. IV.C], the authors analyse the error performance of an AMP decoder to obtain an achievability bound on the minimum E_b/N_0 required for reliable communication in quasi-static fading MACs in the asymptotic regime considered in this chapter. The essence of their AMP analysis is similar to that of Theorem 4: they analyse the error performance of an AMP decoder for i.i.d. (complex) Gaussian design matrices via the fixed points of its state evolution. However, the AMP decoder (and state evolution) used in their work differs from the one we analyse in Theorem 4. Their analysis provides a bound on the achievability of their AMP decoder, whereas Theorem 4 provides the exact achievability of our AMP decoder. Furthermore, their bound can be numerically evaluated for large B (the section size), whereas the result in Theorem 4 is too computationally expensive to numerically evaluate at large B. (In our work, computing the fixed point of the state evolution or the stationary point of the potential function each require a B-dimensional integral.)

Remark 4.2.2. Consider the setting of Theorem 4 and the (high-dimensional) MAP decoder for the linear model (4.2), denoted by $\hat{\boldsymbol{x}}^{\mathrm{MAP}}$. The (non-rigorous) replica analysis in [164–166] shows that the asymptotic user error rate of $\hat{\boldsymbol{x}}^{\mathrm{MAP}}$ can be analysed in terms of probability of decoding error in the single-section Gaussian channel (4.19). Specifically, when $\mathcal{M}(\mu, \sigma^2)$ defined in (4.22) is a singleton,

$$\lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{\mathrm{MAP}} \neq \boldsymbol{x}_\ell\} = P_e(\tau^*), \tag{4.41}$$

where the limit is taken with $\frac{L}{n} = \mu$ held constant and

$$\tau^* = \sigma^2 + \mu \mathcal{M}(\mu, \sigma^2). \tag{4.42}$$

In this setting, and when the sections of the message vector $\boldsymbol{x}$ are i.i.d. $\sim p_1$ (which corresponds to coding with random codebooks), the MAP decoder is the (joint) ML decoder which was analysed in [32, 33]. Therefore, while [32, 33] provided bounds on the asymptotic achievable region of i.i.d. Gaussian codebooks with ML decoding, equation (4.41) provides the exact asymptotic achievable region.[2]

[2] A similar result to (4.41) has been rigorously proven for the (high-dimensional) MMSE decoder for the linear model (4.2), denoted by $\hat{\boldsymbol{x}}^{\mathrm{MMSE}}$: under the same condition on $\mathcal{M}(\mu, \sigma^2)$, we have $\lim_{L\to\infty} \frac{1}{L} \|\hat{\boldsymbol{x}}_\ell^{\mathrm{MMSE}} - \boldsymbol{x}_\ell\|^2 = \mathcal{M}(\mu, \sigma^2)$. This result was proved for the $B = 1$ case ($\boldsymbol{x}$ having i.i.d. entries) in both [84] and [74]. It was

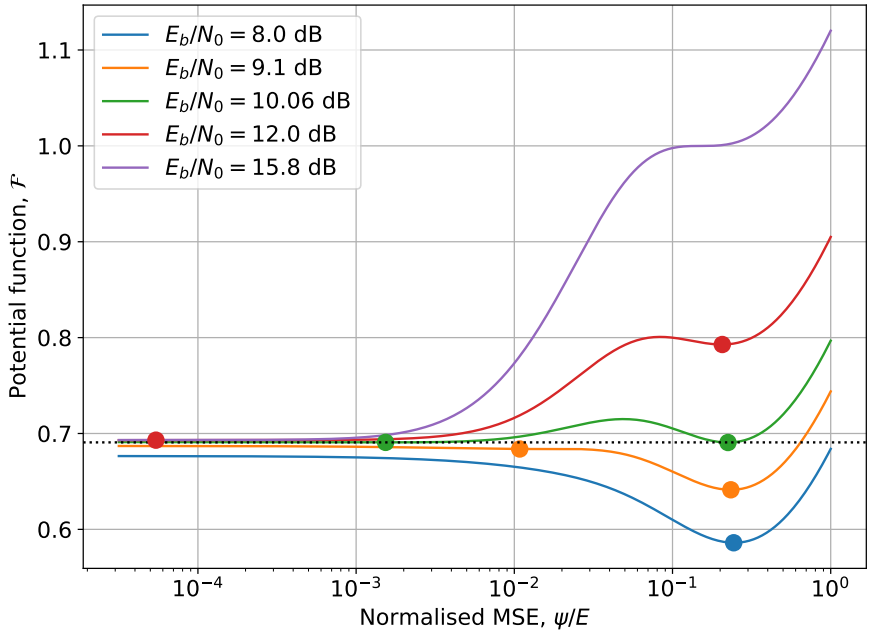


Figure 4.2: The potential function $\mathcal{F}$ as a function of the normalised MSE at several signal-to-noise ratios E_b/N_0, when the user density $\mu = 2$, user payload $= 1$ bit, and $\boldsymbol{X}_{\text{sec}} \sim p_1$ (Example 4.1.1). The solid coloured circles represent the local and global minima of the potential function.

Example 4.2.1. In Fig. 4.2, we show some examples of the potential function $\mathcal{F}$ defined in (4.20) when $\boldsymbol{X}_{\text{sec}} \sim p_1$. Instead of considering the potential function as a function of σ^2 and ψ, we consider it being a function of their normalised forms: $\frac{\sigma^2}{E}$ and $\frac{\psi}{E}$. This allows us to understand the potential function as a function of $\frac{E_b}{N_0}$ since $\frac{\sigma^2}{E} = (2\frac{E_b}{N_0}\log_2 B)^{-1}$. The corresponding potential function is as follows,

$$\mathcal{F}\Big(\mu, \frac{\sigma^2}{E}, \frac{\psi}{E}\Big) = I(\boldsymbol{X}_{\text{sec}}; \boldsymbol{S}_\tau) + \frac{1}{2\mu}\left[\ln\left(1 + \frac{\mu(\psi/E)}{\sigma^2/E}\right) - \frac{\mu(\psi/E)}{(\sigma^2/E) + \mu(\psi/E)}\right], \tag{4.43}$$

where the mutual information term is given in (4.21), which is a function of $\frac{\tau}{E} = \frac{\sigma^2}{E} + \mu\frac{\psi}{E}$. In Fig. 4.2, the potential function is shown as a function of $\frac{\psi}{E}$ when $B = 2$, $\mu = 2.0$, and $\frac{E_b}{N_0}$ takes on a few different values. Numerical integration is used to evaluate the expectation term in (4.21).

Starting from the smallest signal-to-noise ratio $\frac{E_b}{N_0} = 8.0$ dB (blue), the potential function has a unique minimum (stationary point) with corresponding normalised MSE (NMSE) greater than 10^{-1}. Let us denote this NMSE by $\frac{\psi^*}{E}$. From Theorem 4 and Remark 4.2.2, both the AMP and MAP decoder achieve the same asymptotic UER of $P_e(\tau^*)$, where $\frac{\tau^*}{E} = \frac{\sigma^2}{E} + \mu\frac{\psi^*}{E}$ and P_e is defined in (4.26) for $\boldsymbol{X}_{\text{sec}} \sim p_1$. At $\frac{E_b}{N_0} = 9.1$ dB, there are two minima, one global and one local. Since the global minimum and largest stationary point coincide (with corresponding

mentioned in [84] that the result would hold for general $B \geq 1$ if the MSE of the AMP decoder for spatially coupled design matrices can be shown to concentrate around their corresponding state evolution prediction for $B \geq 1$. Such a concentration result has since been shown in [1] for the specific distribution of $p_{\boldsymbol{X}_{\text{sec}}}$ in Example 4.1.1. The proof for more general discrete priors is essentially the same.

NMSE $> 10^{-1}$), the AMP and MAP decoder again achieve the same asymptotic UER. At $\frac{E_b}{N_0} = 10.06$ dB, there are two minima with the same potential function value. Hence, the minimiser of the potential function is not unique. The minima at higher NMSE, which is also the largest stationary point, can be used to calculate the asymptotic UER of the AMP decoder (Theorem 4). For $\frac{E_b}{N_0}$ between 10.06 dB and 15.8 dB (e.g. $\frac{E_b}{N_0} = 12.0$ dB), the potential function has two minima, the global minima at low NMSE and the local minima (also the largest stationary point) at high NMSE. The low and high NMSE value can be used to calculate the asymptotic UER achieved by the MAP and AMP decoder, respectively. For $\frac{E_b}{N_0} \geq 15.8$ dB, there one global minimum at low NMSE ($< 10^{-5}$), which can be used to calculate to the asymptotic UER achieved by both MAP and AMP decoders.

4.2.3 Spatially coupled Gaussian matrices

Theorem 5 (Spatially coupled Gaussian matrices with AMP decoding)**.** *Consider the linear model* (4.2) *with a spatially coupled design matrix* $\boldsymbol{A}$ *constructed using an* (ω, Λ, ρ) *base matrix, and the* L *sections of the message vector* $\boldsymbol{x}$ *i.i.d.* $\sim p_{\boldsymbol{X}_{sec}}$*. Let* $\hat{\boldsymbol{x}}^t$ *be the AMP hard-decision estimate of* $\boldsymbol{x}$ *after iteration* t*, and recall that* $\boldsymbol{\tau}^t \in \mathbb{R}^{\mathsf{C}}$ *is an output of the state evolution* (4.7)–(4.8) *and* $\mathsf{C} = \Lambda$ *in this setting.*

1) For any (ω, Λ, ρ) *base matrix, each entry of* $\boldsymbol{\tau}^t \in \mathbb{R}^{\mathsf{C}}$ *is non-increasing in* t*, and the* c*-th entry converges to a fixed point, denoted by* $\tau_{\mathsf{c}}^{SC\text{-}FP}$*, for* $\mathsf{c} \in [\mathsf{C}]$*.*

2) For any $\epsilon > 0$*, there are constants* $\omega_0 < \infty$*,* $\Lambda_0 < \infty$ *and* $\rho_0 > 0$ *such that, for all* $\omega > \omega_0$*,* $\Lambda > \Lambda_0$ *and* $0 \leq \rho < \rho_0$*, the fixed points* $\{\tau_{\mathsf{c}}^{SC\text{-}FP}\}_{\mathsf{c}\in[\mathsf{C}]}$ *satisfy*

$$\max_{\mathsf{c}\in[\mathsf{C}]} \tau_{\mathsf{c}}^{SC\text{-}FP} \leq \overline{\tau}_{\vartheta} \coloneqq \sigma^2 + \vartheta\mu(\max \mathcal{M}(\vartheta\mu, \sigma^2) + \epsilon), \tag{4.44}$$

where $\vartheta = 1 + (\omega - 1)/\Lambda$*, and the set of potential function minimisers* $\mathcal{M}(\vartheta\mu, \sigma^2)$ *is defined in* (4.22)*.*

3) Fix base matrix parameters $\omega > \omega_0$*,* $\Lambda > \Lambda_0$*, and* $0 < \rho < \rho_0$*. Fix* $\delta > 0$*, and let* T *denote the first iteration for which* $\max_{\mathsf{c}} \tau_{\mathsf{c}}^t \leq \tau_{\mathsf{c}}^{SC\text{-}FP} + \delta$*. Then the user error rate of the AMP decoder after* $T + 1$ *iterations satisfies*

$$\lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{T+1} \neq \boldsymbol{x}_\ell\} \overset{a.s.}{=} \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} P_e(\tau_{\mathsf{c}}^T) \leq P_e(\overline{\tau}_{\vartheta} + \delta), \tag{4.45}$$

where the limit is taken with $\frac{L}{n} = \mu$ *held constant.*

Remark 4.2.3 (Threshold saturation)**.** Theorem 5 shows that the asymptotic user error rate achievable with a suitable spatially coupled Gaussian matrix and AMP decoding is bounded by $P_e(\overline{\tau}_{\vartheta} + \delta)$. If $\Lambda \gg \omega$, we have $\vartheta \to 1$. Therefore, if $\mathcal{M}(\mu, \sigma^2)$ defined in (4.22) is a singleton,

(noting that ϵ in (4.44) can be arbitrarily small) we have

$$\lim_{\omega\to\infty}\lim_{\Lambda\to\infty}\overline{\tau}_{\vartheta}\to\tau^*, \tag{4.46}$$

where τ^* is defined in (4.42). Therefore, in the limit described in (4.46), the asymptotic UER of the spatially coupled scheme with AMP decoding is bounded by $P_e(\tau^*+\delta)$ for any fixed $\delta>0$. This matches the (predicted) asymptotic UER achieved by i.i.d. Gaussian matrices and MAP decoding (Remark 4.2.2).

This phenomenon, where the performance of message passing decoding in a spatially coupled system matches the MAP (or MMSE) decoding performance in the corresponding uncoupled system, has been shown in other applications and is known as *threshold saturation* [14, 15, 83, 134, 139, 140].

Proof of Theorem 5. 1) Consider the spatially coupled state evolution (4.7)–(4.8) as a single line recursion in the vector $\boldsymbol{\gamma}^t\in\mathbb{R}^{\mathsf{R}}$: for $t\geq 0$ and $\mathsf{r}\in[\mathsf{R}]$,

$$\gamma_{\mathsf{r}}^{t+1}=\sum_{\mathsf{c}=1}^{\mathsf{C}}W_{\mathsf{rc}}\,\mathsf{mmse}\Bigg(\sum_{\mathsf{r}'=1}^{\mathsf{R}}W_{\mathsf{r}'\mathsf{c}}\,\frac{1}{\sigma^2+\mu_{\text{inner}}\gamma_{\mathsf{r}'}^t}\Bigg). \tag{4.47}$$

For any base matrix $\boldsymbol{W}$ with non-negative entries (which includes (ω,Λ,ρ) base matrices), the result in [140, Cor. 4.3] shows that each entry of $\boldsymbol{\gamma}^t$ is non-increasing in t and converges to a fixed point; we denote these fixed points by $\{\gamma_{\mathsf{r}}^{\text{SC-FP}}\}_{\mathsf{r}\in[\mathsf{R}]}$. Since the entries of the state evolution parameter $\boldsymbol{\tau}^t$ are non-decreasing in $\{\gamma_{\mathsf{r}}^t\}_{\mathsf{r}\in[\mathsf{R}]}$:

$$\tau_{\mathsf{c}}^t=\Bigg[\sum_{\mathsf{r}=1}^{\mathsf{R}}\frac{W_{\mathsf{rc}}}{\sigma^2+\mu_{\text{inner}}\gamma_{\mathsf{r}}^t}\Bigg]^{-1},\quad\text{for }\mathsf{c}\in[\mathsf{C}], \tag{4.48}$$

we conclude that each entry of $\boldsymbol{\tau}^t$ is also non-increasing in t and converges to a fixed point; these fixed points are denoted $\{\tau_{\mathsf{c}}^{\text{SC-FP}}\}_{\mathsf{c}\in[\mathsf{C}]}$. The arguments used in [140, Cor. 4.3] are similar to those used in the proof of Theorem 4 to show that the uncoupled state evolution parameters converge to fixed points.

2) The result (4.44) is obtained by using the results in [83] on the fixed points of general coupled recursions. We now describe the steps we take to apply the results in [83]. The uncoupled state evolution (4.18) can be written as a single line recursion:

$$\psi^{t+1}=\mathsf{mmse}\Bigg(\frac{1}{\sigma^2+\mu\psi^t}\Bigg). \tag{4.49}$$

The uncoupled recursion in (4.49) and the coupled recursion in (4.47) correspond exactly to [83, Eqs. (27)–(28)] when μ of the uncoupled system is equal to μ_{inner} of the spatially coupled system and $\boldsymbol{W}$ is an $(\omega,\Lambda,\rho=0)$ base matrix. (We will discuss the implications of ρ being a small

positive constant later.) Using the same arguments as in [83, Sec. VI.E], and using the vector I-MMSE relationship [85, Thm. 2], we obtain the following result by applying [83, Thms. 1 and 2].

For $\rho = 0$ and any $\epsilon > 0$, there is an $\omega_0 < \infty$ and $\Lambda_0 < \infty$ such that, for all $\omega > \omega_0$ and $\Lambda > \Lambda_0$, the fixed point of (4.47) satisfies

$$\min \mathcal{M}_1(\mu_{\text{inner}}, \sigma^2) - \epsilon \leq \max_{\mathsf{r} \in [\mathsf{R}]} \gamma_{\mathsf{r}}^{\text{SC-FP}} \leq \max \mathcal{M}_1(\mu_{\text{inner}}, \sigma^2) + \epsilon, \tag{4.50}$$

where

$$\mathcal{M}_1(\mu, \sigma^2) = \left\{ \arg\min_{\psi \in [0,E]} \mathcal{F}_1(\mu, \sigma^2, \psi) \right\}, \tag{4.51}$$

$$\mathcal{F}_1(\mu, \sigma^2, \psi) = 2\Bigg(I\bigg(\boldsymbol{X}_{\text{sec}}; \sqrt{\frac{1}{\sigma^2 + \mu\psi}} \boldsymbol{X}_{\text{sec}} + \boldsymbol{Z} \bigg) - I\bigg(\boldsymbol{X}_{\text{sec}}; \sqrt{\frac{1}{\sigma^2}} \boldsymbol{X}_{\text{sec}} + \boldsymbol{Z} \bigg) + \frac{1}{2\mu} \bigg[\ln\Big(1 + \frac{\mu\psi}{\sigma^2}\Big) - \frac{\mu\psi}{\sigma^2 + \mu\psi} \bigg] \Bigg), \tag{4.52}$$

and where $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$ and $\boldsymbol{Z} \in \mathbb{R}^B$ is a standard Gaussian vector independent of $\boldsymbol{X}_{\text{sec}}$. Since $\mathcal{F}_1(\mu, \sigma^2, \psi)$ and the potential function $\mathcal{F}(\mu, \sigma^2, \psi)$ defined in (4.20) are equivalent after removing constant scaling factors and terms that don't depend on ψ, their minimisers with respect to ψ are identical. Therefore, we can write (4.50) as

$$\min \mathcal{M}(\mu_{\text{inner}}, \sigma^2) - \epsilon \leq \max_{\mathsf{r} \in [\mathsf{R}]} \gamma_{\mathsf{r}}^{\text{SC-FP}} \leq \max \mathcal{M}(\mu_{\text{inner}}, \sigma^2) + \epsilon, \tag{4.53}$$

where $\mathcal{M}(\mu, \sigma^2)$ is the set of minimisers of $\mathcal{F}(\mu, \sigma^2, \psi)$ defined in (4.22).

Now we consider the effect of ρ being a small positive constant on the fixed point of the state evolution. We study this scenario as ρ needs to be lower bounded by a strictly positive constant for the AMP concentration result in (4.45) to hold. First, the $\mathsf{mmse}(\mathsf{snr})$ function defined in (4.9) is a smooth function of snr on $(0, \infty)$ [163, Prop. 7]. Therefore, the right-hand-side of (4.47) is a smooth function of the entries of $\boldsymbol{W}$. Hence, the fixed point of the state evolution recursion (4.47) is a smooth function of ρ. For $\rho \geq 0$, denoting this fixed point by $\{\gamma_{\mathsf{r}}^{\text{SC-FP}}(\rho)\}_{\mathsf{r} \in [\mathsf{R}]}$, and letting

$$\Delta(\rho) := \max_{\mathsf{r} \in [\mathsf{R}]} \left| \gamma_{\mathsf{r}}^{\text{SC-FP}}(\rho) - \gamma_{\mathsf{r}}^{\text{SC-FP}}(0) \right|,$$

we have $\Delta(\rho) \to 0$ as $\rho \to 0$. Consequently, the result for $(\omega, \Lambda, \rho = 0)$ base matrices in (4.53) holds for $(\omega, \Lambda, \rho > 0)$ base matrices with the deviation ϵ replaced by the slightly larger value $\epsilon + \Delta(\rho)$. Equivalently, since $\epsilon > 0$ is arbitrary and $\Delta(\rho)$ is a smooth function with $\Delta(0) = 0$, there exists a $\rho_0 > 0$ such that, for all $\rho < \rho_0$, the result (4.53) holds for $(\omega, \Lambda, \rho > 0)$ base matrices.

We now obtain (4.44) using (4.53). For $\mathsf{c} \in [\mathsf{C}]$, we have

$$\tau_{\mathsf{c}}^{\text{SC-FP}} = \left[\sum_{\mathsf{r}=1}^{\mathsf{R}} \frac{W_{\mathsf{rc}}}{\sigma^2 + \mu_{\text{inner}}\gamma_{\mathsf{r}}^{\text{SC-FP}}}\right]^{-1} \leq \left[\frac{\sum_{\mathsf{r}=1}^{\mathsf{R}} W_{\mathsf{rc}}}{\sigma^2 + \mu_{\text{inner}} \max_{\mathsf{r}' \in [\mathsf{R}]} \gamma_{\mathsf{r}'}^{\text{SC-FP}}}\right]^{-1}$$
$$\leq \sigma^2 + \mu_{\text{inner}}(\max \mathcal{M}(\mu_{\text{inner}}, \sigma^2) + \epsilon),$$

where the last inequality is obtained using the $\sum_{\mathsf{r}=1}^{\mathsf{R}} W_{\mathsf{rc}} = 1$ constraint on base matrices, and the upper bound in (4.53). The result (4.44) follows by recalling from (4.5) that $\mu_{\text{inner}} = \vartheta\mu$, where $\vartheta = 1 + (\omega - 1)/\Lambda$.

3) We now prove (4.45). Consider the input to the AMP hard-decision step in iteration $t+1$, which we denote by $\boldsymbol{s}^t = \boldsymbol{x}^t + (\widetilde{\boldsymbol{Q}}^t \odot \boldsymbol{A})^* \boldsymbol{z}^t$ (see (4.6), (4.14)). For $\ell \in [L]$, we denote by $\boldsymbol{a}_\ell \in \mathbb{R}^B$ the ℓ-th section of a vector $\boldsymbol{a} \in \mathbb{R}^{LB}$.

The results in [100] and [1] imply that for any pseudo-Lipschitz function $\psi : \mathbb{R}^B \times \mathbb{R}^B \to \mathbb{R}$, the following holds almost surely:

$$\lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \psi\left(\boldsymbol{x}_\ell, \boldsymbol{s}_\ell^t\right) = \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} \mathbb{E}\left\{\psi\left(\boldsymbol{X}_{\text{sec}}, \boldsymbol{S}_{\tau_{\mathsf{c}}^t}\right)\right\}, \tag{4.54}$$

where the limit is taken with $L/n = \mu$ held constant, $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$ and $\boldsymbol{S}_{\tau_{\mathsf{c}}^t}$ is given by (4.19). This result was proved in [100] for $B = 1$, and extended in [1] to the setting of sparse regression codes where the specific distribution $p_{\boldsymbol{X}_{\text{sec}}}$ given in Example 4.1.1 (corresponding to random codebooks) is used. The proof for more general discrete distributions is essentially the same.

Then, following the same steps as (4.32)–(4.39) (using (4.54) instead of (4.36) in (4.37)) gives the desired result. □

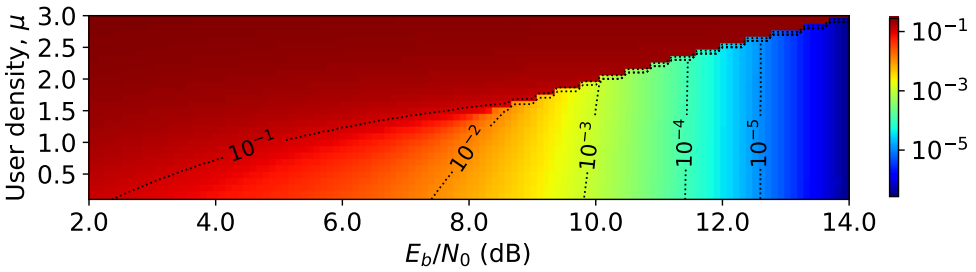
4.2.4 Numerical results

Theorems 4 and 5 give us the asymptotic UER achieved by the AMP decoder when i.i.d. and spatially coupled Gaussian design matrices are used. These results are given in terms of the largest stationary point and (global) minimum of the potential function defined in (4.20). In this section we numerically evaluate these results for certain system parameter choices.

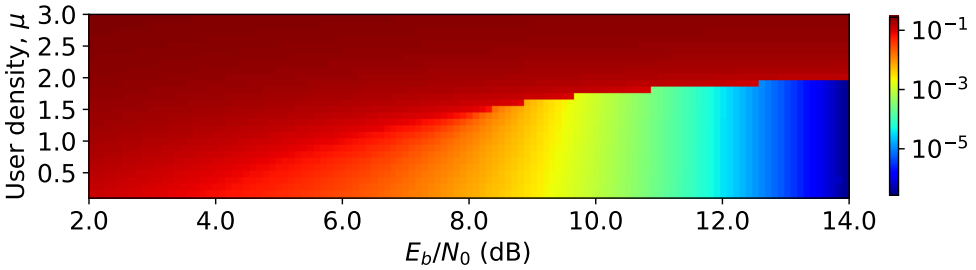
Asymptotic UER heat map for 1 bit user payload

Fig. 4.3 plots the asymptotic UERs achieved by AMP decoding with i.i.d. and spatially coupled Gaussian design matrices as the user density μ and signal-to-noise ratio E_b/N_0 is varied. We consider the setup where each user transmits a payload of 1 bit using a codebook with $B = 2$ Gaussian codewords (see Example 4.1.1).

Specifically, for each (E_b/N_0, μ) point on a 2D grid, we calculate $\sigma^2/E = (2\frac{E_b}{N_0}\log_2 B)^{-1}$ and evaluate the potential function in (4.43) for a list of ψ/E values, using the first expression



(a) Spatially coupled Gaussian matrices with AMP decoding



(b) I.I.D. Gaussian matrices with AMP decoding

Figure 4.3: Heat maps of the asymptotic UER achieved by AMP decoding and either i.i.d. or spatially coupled Gaussian design matrices. The user payload is 1 bit and users encode their information using random codebooks defined by the design matrix.

in (4.21) for the mutual information term.[3] We find the ψ/E values at the global minimum and largest stationary point and use them to calculate $\tau/E = \sigma^2/E + \psi/E$, which is then used to calculate the UER using the first expression in (4.26). Recall that the global minimum corresponds to the asymptotic UER achieved by spatially coupled matrices and AMP decoding in the limit of large coupling parameters described in Remark 4.2.3, and the largest stationary corresponds to that achieved by i.i.d. matrices and AMP decoding. The colour of each grid point in Fig. 4.3 represents the UER value.

In Fig. 4.3, we first observe that the asymptotic UER achieved using both i.i.d. and spatially coupled Gaussian matrices are exactly the same for user densities $\mu < 2.0$. When both μ and $\frac{E_b}{N_0}$ are large (top right section of the figures), using spatially coupled matrices achieves lower UER. Looking at the asymptotic UER heat map of spatially coupled matrices with AMP decoding (Fig. 4.3a), we make the following observations:

1. Fix μ and increase E_b/N_0; the UER decreases as expected. However, the decrease in UER is more gradual at low user densities, whereas at high user densities the UER undergoes a sharp step drop: the UER is high ($> 10^{-1}$) for all values of E_b/N_0 below a certain threshold and then suddenly drops to a low value ($< 10^{-3}$) once E_b/N_0 exceeds that threshold.

[3]The expectation term in the potential function can be evaluated using either numerical integration or Monte Carlo methods.

2. Fix E_b/N_0 and increase μ; the UER increases as expected. However, the increase in UER is more gradual at low E_b/N_0, and at high E_b/N_0 the UER undergoes a sharp step increase: the UER is low ($< 10^{-3}$) and roughly constant for all values of μ below a certain threshold and then suddenly increases to a large value ($> 10^{-1}$) once μ exceeds that threshold.

3. Consider the contour lines of constant UERs (dotted lines). At high UER (10^{-1}–10^{-2}), as the user density μ increases, the signal-to-noise ratio E_b/N_0 also needs to increase to keep the UER constant, which is expected. However, at low UER ($\leq 10^{-4}$), as the user density μ increases, E_b/N_0 may not need to increase to keep the UER constant, it depends on whether the user density is larger than a certain threshold.

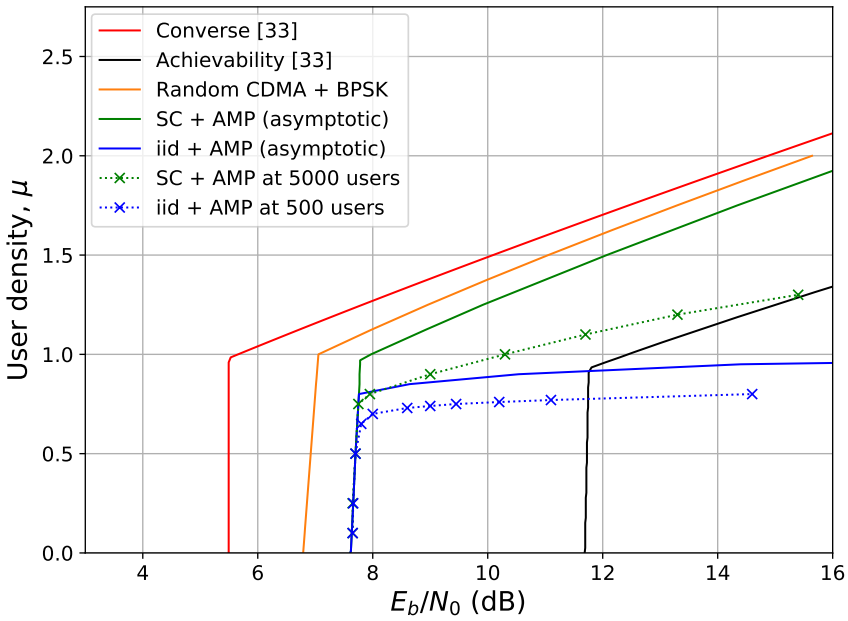
Asymptotic achievable region for fixed payload and target UER

Fig. 4.4 plots the asymptotic achievable region of AMP decoding with i.i.d. and spatially coupled Gaussian codebooks, i.e., when the design matrix $\boldsymbol{A}$ is either i.i.d. or spatially coupled Gaussian, and the sections of the message vector $\boldsymbol{x}$ are drawn i.i.d. from p_1. Specifically, for a list of user densities μ, we plot the minimum E_b/N_0 required by the coding schemes to achieve a UER of less than 10^{-3}, when the user payload is 2 or 8 bits.

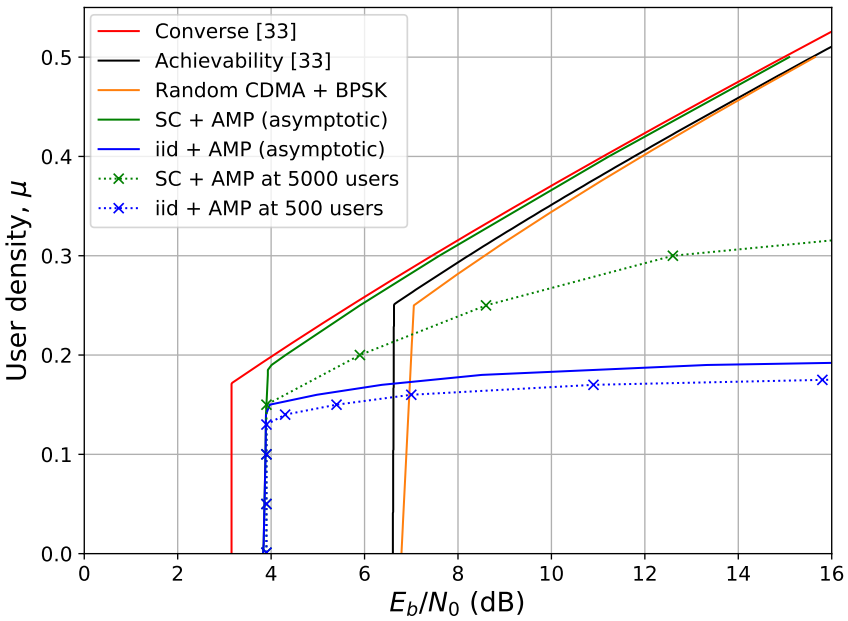
The asymptotic achievable region of the i.i.d. coding scheme is given by Theorem 4 (solid blue lines), and that of spatially coupled coding scheme given by Theorem 5 and Remark 4.2.3 (solid green lines). We observe that the asymptotic achievable region of spatially coupled Gaussian codebooks with efficient AMP decoding is strictly larger than the achievability bound in [33] (black lines), which is based on i.i.d. Gaussian codebooks and ML decoding. Moreover, in Fig. 4.4b where the user payload is 8 bits, it nearly matches the converse bound (red line) for $\mu \geq 0.2$. We observe that gaps between the bounds (converse, achievability, and asymptotic SC + AMP) are much wider when the user payload is 2 bits compared to 8 bits.

At low user densities ($\mu \leq 0.80$ in Fig. 4.4a and $\mu \leq 0.15$ in Fig. 4.4b), we observe that the minimum E_b/N_0 required by the i.i.d. and spatially coupled coding schemes is the same. However, the gap between the achievable regions of the two schemes increases sharply for larger μ. Furthermore, the shape of the solid blue curve suggests that it might be impossible to achieve UER $\leq 10^{-3}$ with i.i.d. Gaussian codebooks and AMP decoding above a certain user density ($\mu \approx 1.0$ in Fig. 4.4a and $\mu \approx 0.2$ in Fig. 4.4b).

We also show the simulated performance of the i.i.d. and spatially coupled coding schemes at 500 and 5000 users, respectively (dotted lines with crosses). For a list of user densities μ, the crosses show the minimum E_b/N_0 at which the coding scheme achieves an average UER less than 10^{-3} (averaged over many independent trials). Discrete cosine transform (DCT) based design matrices were used to reduce decoding complexity and memory usage (see Section 2.5.1 for more details). The error rates obtained using DCT and Gaussian matrices are similar for large matrix sizes. The simulations for the spatially coupled coding scheme used (ω, $\Lambda = 50$,



(a) 2 bits ($B = 2^2$)



(b) 8 bits ($B = 2^8$)

Figure 4.4: Achievable regions for massive multiple access at a user payload of 2 bits and 8 bits, when the maximum tolerated UER is 10^{-3}. Users encode their information using random codebooks defined by the design matrix.

$\rho = 0$) base matrices. The coupling width ω was optimised for each user density μ (see Table 4.1).[4]

[4]When spatially coupled design matrices are used, the actual user density differs slightly from the design user density μ (which we plot) because a rounding procedure is carried out to ensure that the design matrix is split into blocks with an integer number of rows and columns. For example, at user density $\mu = 1.0$, $L = 5000$ users, coupling length $\Lambda = 50$ and coupling width $\omega = 5$, the codelength is $n = 5000$ and the base matrix has $\mathsf{R} = 54$ rows and $\mathsf{C} = 50$ columns. Therefore, each block of the design matrix has $5000/54 = 92.59 \approx 93$ rows and

Table 4.1: Optimised coupling width values used in Figs. 4.4a and 4.4b.

	Fig. 4.4a					Fig. 4.4b				
μ	0.9	1.00	1.1	1.2	1.3	0.15	0.20	0.25	0.30	0.33
ω	5	5	6	6	7	5	5	6	11	14

We observe that for both i.i.d. and spatially coupled coding schemes, the finite user and asymptotic curves match at low user densities. For the i.i.d. coding scheme (blue), although a gap between the two curves appears above a certain user density threshold ($\mu = 0.5$ in Fig. 4.4a and $\mu = 0.13$ in Fig. 4.4b), their overall shape remains similar. For the spatially coupled scheme (green), the gap between the two curves appears above a user density threshold slightly higher than that of the i.i.d. coding scheme, and increases with μ. This gap is a finite length effect, due to the relatively small values of base matrix parameters.

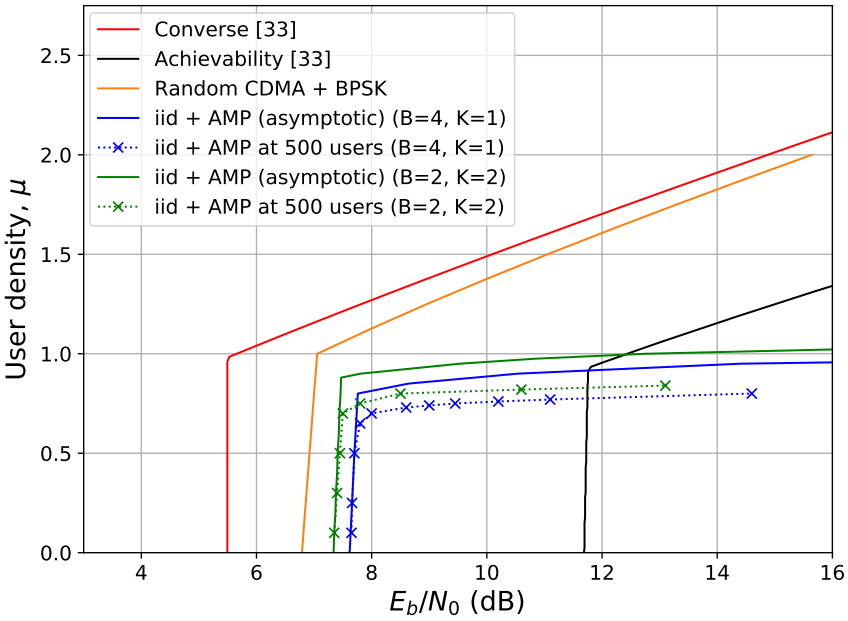
Table 4.1 shows the values of the optimised coupling widths ω used in Figs. 4.4a and 4.4b, for user densities above the threshold at which the spatially coupled coding scheme has an advantage over the i.i.d. coding scheme. At user densities lower than this threshold, a range of coupling widths (including the uncoupled case $\omega = 1$) achieve similar UERs. We see that the optimal coupling width increases with the user density.

In Fig. 4.4, we also plot the asymptotic achievable region of the coding scheme based on random code-division multiple access (CDMA) with binary phase shift keying (BPSK) modulation and MAP decoding for comparison (solid orange lines). Recall that this encoding scheme corresponds to a random linear system (4.2) with an i.i.d. Gaussian design matrix and a message vector with i.i.d. entires drawn uniformly from $\{\pm\sqrt{E}\}$ (see Example 4.1.2). The asymptotic achievable region of the MAP decoder is predicted using the result from the replica analysis (see Remark 4.2.2). Since each user can only encode 1 bit per n transmissions in this coding scheme, encoding 2 bits and 8 bits would require $2n$ and $8n$ transmissions, which corresponds to a reduction in the user density $\mu = L/n$ by factors of 2 and 8, respectively.

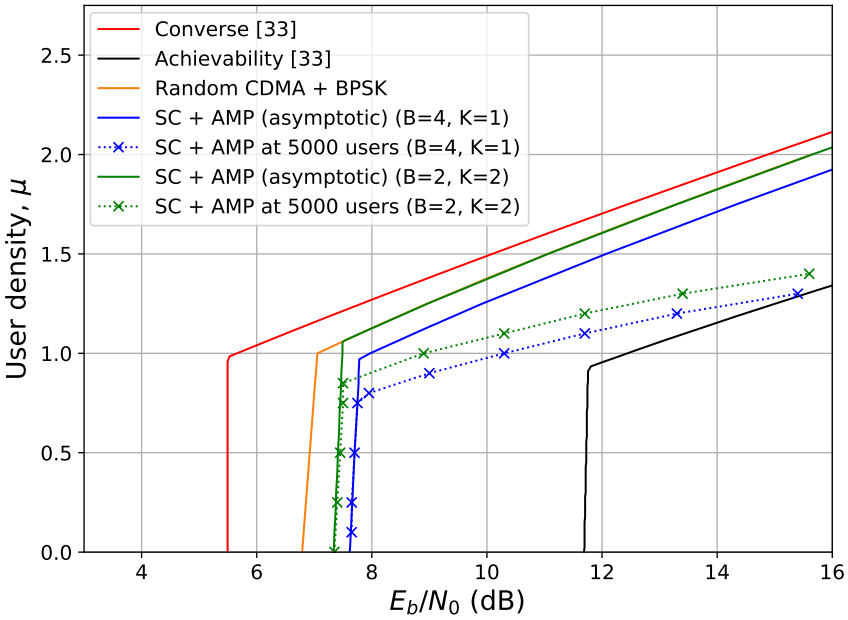
The effect of modulation In Fig. 4.5, we investigate how the achievable region changes when modulation is introduced to the encoding process. We consider encoding with Gaussian codebooks and binary modulation, i.e., in addition to encoding $\log_2 B$ bits in the choice of one-of-B Gaussian codewords, an additional bit is encoded in the sign of the chosen codeword (see Example 4.1.2).

In Fig. 4.5 we plot the asymptotic achievable region when the user payloads is 2 bits and the maximum tolerated UER is 10^{-3} (same as in Fig. 4.4a). The unmodulated scheme ($B = 4$, $K = 1$) will encode 2 bits by choosing one of $B = 4$ codewords, whereas the binary modulated scheme ($B = 2$, $K = 2$) would encode 2 bits by using 1 bit to choose one of $B = 2$ codewords and the other bit to choose one of $K = 2$ signs.

$5000/50 = 100$ columns, and the actual user density is $5000/(93 \times 54) \approx 0.9956$.



(a) I.I.D. design matrix + AMP



(b) Spatially coupled design matrix + AMP

Figure 4.5: Achievable regions for massive multiple access at a user payload of 2 bits, when the maximum tolerated UER is 10^{-3}. The $K = 1$ case corresponds to encoding with random codebooks (defined by the design matrix) and the $K = 2$ case corresponds to encoding with random codebooks and binary modulation.

In Figs. 4.5a and 4.5b, both the asymptotic and finite user achievability region of the binary modulated scheme (green lines) are noticeably larger than that of the unmodulated scheme (blue lines), for both i.i.d. and spatially coupled design matrices. We did not plot the achievable region of the binary modulated scheme when the user payload is 8 bits because its difference with the

unmodulated scheme (Fig. 4.4b) is less than 0.1 dBs at all user densities.

In Fig. 4.5b, the spatially coupled design matrices used in finite user simulations were constructed using (ω, $\Lambda = 50$, $\rho = 0$) base matrices. The optimal coupling width for both the binary modulated and unmodulated schemes is the same at each user density (see Table 4.1). (The optimal coupling width for the binary modulated scheme at $\mu = 1.4$ is 9 — it continues to increase with the user density.)

4.3 Large user payloads

When coding with random codebooks, i.e., when the sections of $\boldsymbol{x}$ are drawn i.i.d. from p_1, the section size B increases exponentially with the user payload ($\log_2 B$ bits). For large B it is infeasible to evaluate the potential function (4.20), and the potential function is needed to compute the asymptotic UER bounds in Theorems 4 and 5 (see (4.29) and (4.45)). In this section we bound the asymptotic UER achieved by i.i.d. and spatially coupled Gaussian codebooks with AMP decoding, when the user payload is large. Both results (Theorems 6 and 7) utilise the following Lemma.

Lemma 4.3.1 (Asymptotic UER bound)**.** *Consider the setting of either Theorem 4 or 5, and take the distribution $p_{\boldsymbol{X}_{sec}}$ to be p_1. Let $\hat{\boldsymbol{x}}^t$ be the AMP hard-decision estimate of $\boldsymbol{x}$ after iteration t, and recall that $\boldsymbol{\psi}^t \in \mathbb{R}^{\mathsf{C}}$ is an output of the state evolution* (4.7)–(4.8)*. Then we have that the following limit exists almost surely and satisfies:*

$$\frac{1}{2\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} \frac{\psi_{\mathsf{c}}^t}{E} \leq \lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^t \neq \boldsymbol{x}_\ell\} \leq \frac{4}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} \frac{\psi_{\mathsf{c}}^t}{E}, \tag{4.55}$$

where the limit is taken with $\frac{L}{n} = \mu$ held constant. Recall that $\mathsf{C} = 1$ when the design matrix $\boldsymbol{A}$ has i.i.d. Gaussian entries $\mathcal{N}(0, \frac{1}{n})$.

Proof. The existence of the limit in (4.55) is shown in Theorems 4 and 5. For $\ell \in [L]$, we denote by $\boldsymbol{a}_\ell \in \mathbb{R}^B$ the ℓ-th section of a vector $\boldsymbol{a} \in \mathbb{R}^{LB}$. Let $\boldsymbol{x}^{t+1}$ be the AMP estimate of $\boldsymbol{x}$ after iteration $t+1$ (defined in (4.13)).

It was proved in [1, Thm. 2] that the MSE of the AMP decoder after iteration $t \geq 0$ converges almost surely to the following limit:

$$\lim_{L\to\infty} \frac{\|\boldsymbol{x}^{t+1} - \boldsymbol{x}\|^2}{L} = \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} \psi_{\mathsf{c}}^{t+1} = \frac{1}{\mathsf{C}} \sum_{\mathsf{c}=1}^{\mathsf{C}} \mathbb{E}\Big\{ \big\| \boldsymbol{X}_{\text{sec}} - \mathbb{E}[\boldsymbol{X}_{\text{sec}} \mid \boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\, \boldsymbol{Z}] \big\|^2 \Big\}, \tag{4.56}$$

where $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$ and $\boldsymbol{Z}$ is a standard Gaussian vector independent of $\boldsymbol{X}_{\text{sec}}$. The last equality in (4.56) follows from the definition of ψ_{c}^t in (4.8)–(4.9). Moreover, using the result (4.45) (noting

that the first equality in (4.45) holds for $t \geq 0$), we have

$$\lim_{L\to\infty} \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{t+1} \neq \boldsymbol{x}_\ell\} = \frac{1}{\mathsf{C}}\sum_{\mathsf{c}=1}^{\mathsf{C}} P_e(\tau_{\mathsf{c}}^t) = \frac{1}{\mathsf{C}}\sum_{\mathsf{c}=1}^{\mathsf{C}} \mathbb{E}\Big[\mathbb{1}\Big\{\boldsymbol{X}_{\text{sec}} \neq \hat{\boldsymbol{X}}(\boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\boldsymbol{Z})\Big\}\Big]$$
$$= \frac{1}{2E}\frac{1}{\mathsf{C}}\sum_{\mathsf{c}=1}^{\mathsf{C}} \mathbb{E}\left[\|\boldsymbol{X}_{\text{sec}} - \hat{\boldsymbol{X}}(\boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\boldsymbol{Z})\|^2\right]. \tag{4.57}$$

Here $\hat{\boldsymbol{X}}(\cdot)$ is the MAP estimator of $\boldsymbol{X}_{\text{sec}}$, i.e.,

$$\hat{\boldsymbol{X}}(\boldsymbol{s}) = \arg\max_{\boldsymbol{x}'} \mathbb{P}\Big(\boldsymbol{X}_{\text{sec}} = \boldsymbol{x}' \mid \boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\,\boldsymbol{Z} = \boldsymbol{s}\Big).$$

The second equality in (4.57) follows from (4.25), noting that $P_e(\tau_{\mathsf{c}}^t) = \mathbb{P}(\boldsymbol{X}_{\text{sec}} \neq \hat{\boldsymbol{X}}(\boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\boldsymbol{Z}))$. The third equality is obtained by noticing that for $\boldsymbol{X}_{\text{sec}} \sim p_1$, the squared error of the MAP estimator $\hat{\boldsymbol{X}}(\boldsymbol{s})$ (defined in (4.15)) satisfies

$$\|\hat{\boldsymbol{X}}(\boldsymbol{s}) - \boldsymbol{X}_{\text{sec}}\|^2 = 2E\,\mathbb{1}\{\hat{\boldsymbol{X}}(\boldsymbol{s}) \neq \boldsymbol{X}_{\text{sec}}\}. \tag{4.58}$$

Among estimators of $\boldsymbol{X}_{\text{sec}}$ from $\boldsymbol{S} = \boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\,\boldsymbol{Z}$, the expected squared loss is minimised by the conditional expectation. Therefore, for $\mathsf{c} \in [\mathsf{C}]$,

$$\mathbb{E}\big\{\big\|\boldsymbol{X}_{\text{sec}} - \mathbb{E}[\boldsymbol{X}_{\text{sec}} \mid \boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\,\boldsymbol{Z}]\big\|^2\big\} \leq \mathbb{E}\big\{\|\boldsymbol{X}_{\text{sec}} - \hat{\boldsymbol{X}}(\boldsymbol{X}_{\text{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\boldsymbol{Z})\|^2\big\}. \tag{4.59}$$

Using (4.59) to compare the limits in (4.56) and (4.57), we obtain first inequality in (4.55).

To prove the second inequality in (4.55), we first notice that for the prior p_1, the hard-decision estimator $\hat{\boldsymbol{x}}_\ell^{t+1}$ defined in (4.15) can equivalently be written as follows. For $j \in \text{sec}(\ell)$:

$$\hat{x}_j^{t+1} = \begin{cases} \sqrt{E} & \text{if } x_j^{t+1} > x_i^{t+1} \text{ for all } i \in \text{sec}(\ell)\backslash j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.60}$$

Here $\boldsymbol{x}^{t+1}$ is the AMP estimate computed according to (4.6) and (4.13).

Let $j^* \in \text{sec}(\ell)$ denote the index of the unique non-zero entry of $\boldsymbol{x}$ in section $\ell \in [L]$, i.e., $x_{j*} = \sqrt{E}$. From (4.13), we note that the sum of the entries in each section of $\boldsymbol{x}^{t+1}$ equals $\sqrt{E}$. The decision rule (4.60) then implies that $\boldsymbol{x}_{j*}^{t+1}$ is less than or equal to $\sqrt{E}/2$ whenever $\hat{\boldsymbol{x}}_\ell^{t+1} \neq \boldsymbol{x}_\ell$. Therefore,

$$\hat{\boldsymbol{x}}_\ell^{t+1} \neq \boldsymbol{x}_\ell \quad \text{implies} \quad \|\boldsymbol{x}_\ell^{t+1} - \boldsymbol{x}_\ell\|^2 \geq E/4. \tag{4.61}$$

Therefore,

$$\frac{1}{L}\sum_{\ell=1}^{L}\mathbb{1}\{\hat{\boldsymbol{x}}_\ell^{t+1}\neq\boldsymbol{x}_\ell\}\leq\frac{1}{L}\sum_{\ell=1}^{L}\frac{4\|\boldsymbol{x}_\ell^{t+1}-\boldsymbol{x}_\ell\|^2}{E}. \tag{4.62}$$

Combining (4.62) with (4.56) yields the second inequality in (4.55). □

4.3.1 I.I.D. Gaussian codebooks

Theorem 6 (AMP decoding of i.i.d. Gaussian codebooks at large user payloads)**.** *Consider the setting of Theorem 4 and take the distribution $p_{\boldsymbol{X}_{sec}}$ to be p_1. The user error rate of the AMP decoder after its first iteration exhibits the following phase transition for sufficiently large payloads of $\log_2 B$ bits.*

1) For any $\delta\in(0,\frac{1}{2})$, let $f_{B,\delta}\coloneqq\frac{B^{-k\delta^2}}{\delta\sqrt{\ln B}}$ where k is a positive constant. If

$$\mu\log_2 B<\frac{1}{2}\left(\frac{1}{(1+\frac{\delta}{2})\ln 2}-\frac{1}{E_b/N_0}\right), \tag{4.63}$$

then $\lim_{L\to\infty}\frac{1}{L}\sum_{\ell=1}^{L}\mathbb{1}\{\hat{\boldsymbol{x}}_\ell^1\neq\boldsymbol{x}_\ell\}\leq f_{B,\delta}$.

2) For any $\tilde{\delta}\in(0,1)$, let $g_{B,\tilde{\delta}}\coloneqq B^{-k_1\tilde{\delta}^2}$ where k_1 is a positive constant. If

$$\mu\log_2 B>\frac{1}{2(1-g_{B,\tilde{\delta}})}\left(\frac{1}{(1-\frac{\tilde{\delta}}{2})\ln 2}-\frac{1}{E_b/N_0}\right), \tag{4.64}$$

then $\lim_{L\to\infty}\frac{1}{L}\sum_{\ell=1}^{L}\mathbb{1}\{\hat{\boldsymbol{x}}_\ell^t\neq\boldsymbol{x}_\ell\}\geq(1-g_{B,\tilde{\delta}})/2$ for all $t\geq 1$.

In both statements, the limits exist and are taken with $\frac{L}{n}=\mu$ held constant.

Proof. From Lemma 4.3.1, we know that the asymptotic UER after each iteration $t\geq 1$ satisfies

$$\frac{\psi^t}{2E}\leq\lim_{L\to\infty}\frac{1}{L}\sum_{\ell=1}^{L}\mathbb{1}\{\hat{\boldsymbol{x}}_\ell^t\neq\boldsymbol{x}_\ell\}\leq\frac{4\psi^t}{E}, \tag{4.65}$$

where ψ^t is the output of the state evolution (4.18) after the t-th iteration.

Using state evolution variables τ^t and ψ^t defined in (4.18), let

$$\nu^t\coloneqq\frac{E}{\tau^t\ln B}=\frac{E}{(\sigma^2+\mu\psi^t)\ln B}. \tag{4.66}$$

From [1, Lem. 4.1] we know that for sufficiently large B and any $\delta\in(0,\frac{1}{2})$, $\tilde{\delta}\in(0,1)$, we have

$$(1-g_{B,\tilde{\delta}})\mathbb{1}\{\nu^t<2-\tilde{\delta}\}<\frac{\psi^{t+1}}{E}\leq 1-(1-f_{B,\delta})\mathbb{1}\{\nu^t>2+\delta\}, \tag{4.67}$$

where $g_{B,\tilde{\delta}}$, $f_{B,\delta}$ are defined in the theorem statement. Using (4.66) in (4.67) and recalling that

$E = E_b \log_2 B$, $\sigma^2 = N_0/2$, we obtain

$$\frac{\psi^{t+1}}{E} \begin{cases} \leq f_{B,\delta} & \text{if } \frac{\psi^t}{E} < \frac{\frac{2}{2+\delta} - \frac{\ln 2}{E_b/N_0}}{2\mu \ln B}, \\ > (1 - g_{B,\tilde{\delta}}) & \text{if } \frac{\psi^t}{E} > \frac{\frac{2}{2-\tilde{\delta}} - \frac{\ln 2}{E_b/N_0}}{2\mu \ln B}. \end{cases} \tag{4.68}$$

We obtain the first part of the theorem by substituting the initial condition $\psi^0 = E$ into the first condition in (4.68), and using a different positive constant in the definition of $f_{B,\delta}$ to account for the factor of 4 in the upper bound in (4.65).

We prove the second statement of the theorem by showing that under (4.64), $\frac{\psi^t}{E} > (1 - g_{B,\tilde{\delta}})$ for all $t \geq 1$. Combining this with the lower bound in (4.65) then yields the required result.

Noting that $\psi^0/E = 1$, assume towards induction that $\frac{\psi^t}{E} > (1 - g_{B,\tilde{\delta}})$ for some $t \geq 0$. Then, from (4.66) we have

$$\nu^t < \frac{1}{(\sigma^2/E \, + \, \mu(1 - g_{B,\tilde{\delta}})) \ln B} = \frac{1}{\ln 2/(2E_b/N_0) \, + \, \mu \log_2 B(1 - g_{B,\tilde{\delta}}) \ln 2}. \tag{4.69}$$

Therefore a sufficient condition for $\nu^t < (2 - \tilde{\delta})$ is

$$\left(\frac{N_0}{2E_b} \, + \, \mu \log_2 B(1 - g_{B,\tilde{\delta}}) \right) \ln 2 > \frac{1}{2 - \tilde{\delta}}. \tag{4.70}$$

Rearranging, we obtain the condition in (4.64). From (4.67) we see that under this condition, $\psi^{t+1}/E > (1 - g_{B,\tilde{\delta}})$. This completes the proof of the induction step, and hence the theorem. □

Remark 4.3.1. From (4.63) and (4.64), we see that for any fixed values of μ and E_b/N_0, the asymptotic UER of AMP decoding is lower bounded by a value that approaches $1/2$ with growing B. Therefore, the interesting regime for large user payloads is when the spectral efficiency

$$S := \mu \log_2 B = \frac{L \log_2 B}{n} \quad \text{bits/transmission}, \tag{4.71}$$

is of constant order. (The spectral efficiency is the total number of bits sent by all the users per channel use.) Theorem 6 can be extended to the asymptotic regime where $L, n, \log_2 B$ all tend to infinity with the spectral efficiency held constant. In this case, the user error rate of the AMP decoder exhibits the following phase transition in this large system limit:

$$\lim_{L,B,n \to \infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^1 \neq \boldsymbol{x}_\ell\} \begin{cases} = 0 & \text{if } S < S_{\text{BP}}, \\ \geq \frac{1}{2} & \text{otherwise}, \end{cases} \tag{4.72}$$

where S_{BP} is the belief propagation (BP) threshold

$$S_{\mathrm{BP}} \coloneqq \frac{1}{2}\left(\frac{1}{\ln 2} - \frac{1}{E_b/N_0}\right). \tag{4.73}$$

From (4.72), we see that positive spectral efficiencies are achievable in this large system setting using i.i.d. Gaussian codebooks and AMP decoding if and only if $E_b/N_0 > \ln 2$.

4.3.2 Spatially coupled Gaussian codebooks

From Remark 4.3.1, we see that for large user payloads and spectral efficiencies less than the BP threshold S_{BP}, one does not require spatial coupling for reliable AMP decoding. The following result shows that any spectral efficiency above S_{BP} and below the converse can be achieved using spatially coupled Gaussian codebooks and AMP decoding.

Theorem 7 (AMP decoding of spatially coupled Gaussian codebooks at large user payloads)**.** *Consider the setting of Theorem 5 and take the distribution $p_{\boldsymbol{X}_{sec}}$ to be p_1. Let $\vartheta = 1 + \frac{\omega-1}{\Lambda}$, $\mu = \frac{L}{n}$, $\mathsf{SNR} = \frac{2E_b}{N_0}\mu \log_2 B$, and define*

$$\Delta \coloneqq \frac{1}{2\vartheta}\ln(1 + \vartheta\,\mathsf{SNR}) - \mu \ln B, \tag{4.74}$$

$$\omega^* \coloneqq \Big(\frac{\vartheta\,\mathsf{SNR}^2}{1 + \vartheta\,\mathsf{SNR}}\Big)\frac{1}{\Delta}, \tag{4.75}$$

$$\rho^* \coloneqq \min\Big\{\frac{\Delta}{3\,\mathsf{SNR}}, \frac{1}{2}\Big\}. \tag{4.76}$$

Let δ be an arbitrary constant in $(0, \min\{\frac{\Delta}{2\mu \ln B}, \frac{1}{2}\})$ and S_{opt} be the positive solution to

$$S_{opt} = \frac{1}{2}\log_2\left(1 + S_{opt}\frac{2E_b}{N_0}\right). \tag{4.77}$$

1) If the spectral efficiency satisfies

$$\frac{1}{\vartheta}S_{BP} \;\le\; \mu \log_2 B \;<\; \frac{1}{\vartheta}S_{opt}, \tag{4.78}$$

and the base matrix parameters satsify $\omega > \omega^*$ and $0 < \rho \le \rho^*$, then, for $t \ge 1$ and $\mathsf{c} \le \max\{\frac{\omega t}{\omega^*}, \lceil\frac{\Lambda}{2}\rceil\}$, we have

$$\psi_{\mathsf{c}}^t = \psi_{\Lambda-\mathsf{c}+1}^t \le E\,h_{B,\delta} \tag{4.79}$$

for sufficiently large B, where $E = E_b \log_2 B$, $h_{B,\delta} \coloneqq \frac{B^{-k_2\delta^2}}{\delta\sqrt{\ln B}}$ and k_2 is a positive constant.

2) Let T denote the first iteration for which $\max_{\mathsf{c}} \psi_{\mathsf{c}}^t \le E\,h_{B,\delta}$. Then we have

$$T \;\le\; \left\lceil \frac{\Lambda\omega^\star}{2\omega} \right\rceil, \tag{4.80}$$

and the user error rate of the AMP decoder after T iterations satisfies

$$\lim_{L\to\infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\{\hat{\boldsymbol{x}}_\ell^T \neq \boldsymbol{x}_\ell\} \leq 4\, h_{B,\delta}, \tag{4.81}$$

where the limit is taken with $\frac{L}{n} = \mu$ held constant.

Proof. The first part of Theorem 7 is a direct application of Proposition 2.3.1, which is the state evolution analysis of spatially coupled sparse regression codes for channel coding over the (single user) AWGN channel. The main change of variables required is that the signal-to-noise ratio in the AWGN channel is replaced by $\mathsf{SNR} = \frac{L(E/n)}{\sigma^2} = \frac{2E_b}{N_0}\mu \log_2 B$. Another change is that the AWGN rate $R = \frac{L \ln B}{n}$ is replaced by $\mu \ln B$. In the second part of Theorem 7, the result in (4.80) comes from (2.44)–(2.45), and the result in (4.81) is a direct application of the upper bound in Lemma 4.3.1. □

Remark 4.3.2. A positive solution to (4.77) exists if and only if $E_b/N_0 > \ln 2$.

Remark 4.3.3 (Parameter choice)**.** Consider spectral efficiency $S = \mu \log_2 B$ bits/transmission. For any spectral efficiency $S < S_{\text{opt}}$, or equivalently any $\frac{E_b}{N_0} > \frac{2^{2S}-1}{2S}$ (which matches the converse bound in [33] with $B \to \infty$ and the target PUPE $\epsilon \to 0$), we can choose design parameters as follows to guarantee that the AMP decoder achieves a small UER at large user payloads.

1) If $S < S_{\text{BP}}$, or equivalently $\frac{E_b}{N_0} > (\frac{1}{\ln 2} - 2S)^{-1}$ for $S < \frac{1}{2\ln 2}$, then using i.i.d. Gaussian codebooks guarantees that the UER is bounded by a small constant at large payloads (Thm. 6).

2) If $S_{\text{BP}} \leq S < S_{\text{opt}}$, we can choose the base matrix parameters ω and Λ as follows to satisfy the conditions of Theorem 7. Let $\vartheta_0 = S_{\text{opt}}/S$, first choose $\omega > \omega^\star(\vartheta_0)$ (with ϑ in (4.75) replaced by ϑ_0). Then choose Λ large enough that $\vartheta = 1 + \frac{\omega-1}{\Lambda} \leq \vartheta_0$. This ensures that $S < S_{\text{opt}}/\vartheta$ and $\omega > \omega^\star(\vartheta)$.

Remark 4.3.4. Theorem 7 can be extended to the setting where $L, n, \log_2 B$ all tend to infinity with the spectral efficiency $S = L \log_2 B/n$ held constant (see Remark 4.3.1). In this asymptotic regime, the result states that for any $S_{\text{BP}} \leq S < S_{\text{opt}}$, the UER with AMP decoding converges almost surely to 0.

4.3.3 Numerical results

Fig. 4.6 shows the achievable regions of i.i.d. and spatially coupled Gaussian codebooks with AMP decoding, in the large system limit of $L, n, \log_2 B$ all tending to infinity with the spectral efficiency $S = L \log_2 B/n$ held constant (Remarks 4.3.1 and 4.3.4). The dashed black line is the achievable region for i.i.d. Gaussian codebooks and the solid black line for spatially coupled Gaussian codebooks. From (4.73), we note that i.i.d. Gaussian codebooks with AMP decoding

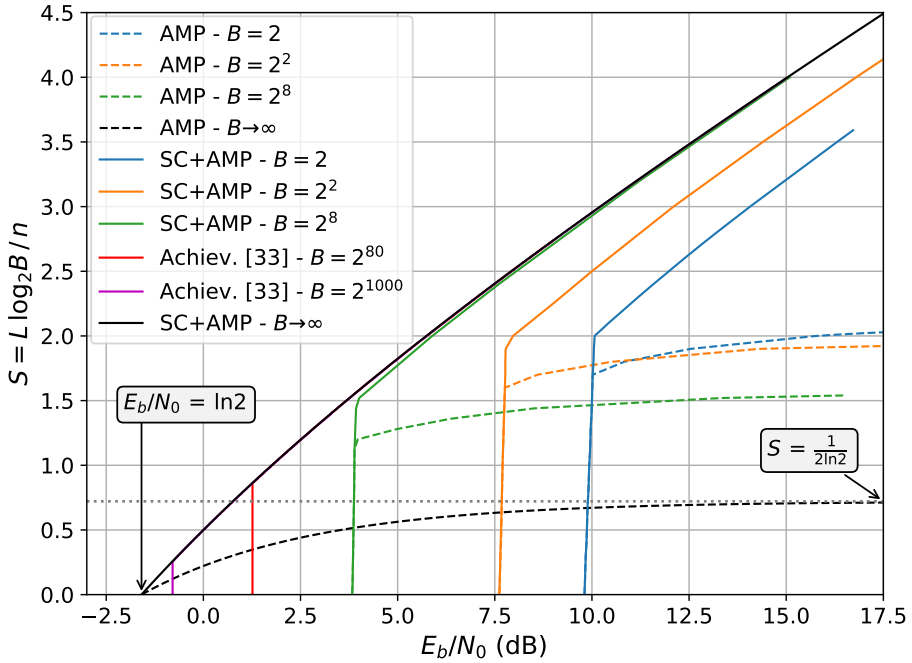


Figure 4.6: Achievable regions for massive multiple access at different user payloads ($\log_2 B$ bits) using either i.i.d. or spatially coupled Gaussian codebooks with AMP decoding. The results at finite B show the minimum E_b/N_0 required to achieve UER $\leq 10^{-3}$.

cannot achieve spectral efficiencies $S \geq \frac{1}{2 \ln 2} \approx 0.7213$ (asymptote of the dashed black line). We also note that the solid black line matches the converse bound in [33] with $B \to \infty$ and the target PUPE $\epsilon \to 0$.

The solid and dashed black lines split Fig. 4.6 into three distinct regions, which are sometimes referred to as the easy, hard, and impossible regions of inference in statistical physics [136]:

1. Below dashed black line: achievable with i.i.d. Gaussian codebooks and AMP decoding.

2. Between solid and dashed black lines: achievable with spatially coupled Gaussian codebooks and AMP decoding, or with i.i.d. Gaussian codebooks and MAP decoding.

3. Above solid black line: not achievable by any scheme.

In Fig. 4.6, we also plot the achievable regions of i.i.d. Gaussian codebooks (dashed) and spatially coupled codebooks (solid) with AMP decoding at several finite payloads $\log_2 B$ (with $L, n \to \infty$ and the user density μ held constant). For $B = 2, 2^2, 2^8$, we use the same setup as Fig. 4.4, and find the smallest E_b/N_0 such that the coding scheme achieves UER $\leq 10^{-3}$. For $B = 2^{80}, 2^{1000}$, it is computationally infeasible to evaluate the potential function (4.20), so we plot the achievability bound from [33] (red and purple curves).

For spatially coupled Gaussian codebooks with AMP decoding, the achievable region gets larger as the user payload increases, but at high spectral efficiencies (e.g. $S > 1.5$), the improvement is insignificant after roughly $\log_2 B = 8$ bits. Therefore, it is possible to communicate reliably at high spectral efficiencies with near-minimal E_b/N_0 even when the user payload is

finite. For i.i.d. Gaussian codebooks with AMP decoding, there is a trade-off in the achievable region as the user payload increases: a lower E_b/N_0 is required to communicate reliably at low spectral efficiencies, but the maximum achievable spectral efficiency decreases.

4.3.4 Implementation

When the user payload ($\log_2 B$ bits) is large, the computational complexity of the AMP decoder is too high for practical use even when DCT based codebooks are used. In this section we discuss how simple modifications to the encoding scheme discussed thus far can reduce the computational complexity, making it feasible for large user payloads. Furthermore, we discuss how introducing modulation to the encoding scheme can further reduce the complexity and improve the achievable region.

The idea to reduce the complexity is simple: instead of encoding the each user's message using a single large codebook, one can encode it using several smaller sized codebooks. This can be implemented in one of the following two ways.

1. Transmit using a smaller codebook multiple times. For example, a user payload of 80 bits can be transmitted by using a codebook of size $B = 2^8$ ten times. The codebooks can be based on either i.i.d. or spatially coupled Gaussian matrices, and the messages can be decoded using an AMP decoder.

2. Transmit by superposition coding of codewords from multiple (different) smaller codebooks. For example, a user payload of 80 bits can be transmitted with each user encoding their message with 10 codebooks of size $B = 2^8$ each. The codewords from the 10 codebooks are summed together to form the final user codeword. This method is equivalent to each user encoding their message with a sparse regression code (introduced in Section 1.3). If we consider the linear model (4.2) where the message vector $\boldsymbol{x}$ has unmodulated sections of length B (Example 4.1.1), then this method is equivalent to each user encoding their message using 10 sections of size $B = 2^8$ each (instead of 1 section of size $B = 2^{80}$).

We now consider the achievability regions of above methods. Compared to the coding scheme where the user payload is 8 bits and each user encodes with a single codebook of size $B = 2^8$, the first method effectively increases the code length by a factor of 10 due to repeated transmissions, and the second method effectively reduces the number of users by a factor of 10 due to each user using 10 codebooks (sections). Therefore, the achievable region obtained using these two methods in terms of the user density $\mu = \frac{\text{number of users}}{\text{code length}}$ versus E_b/N_0 trade-off, is the same as that obtained when the user payload is 8 bits and each user encodes with a single codebook of size $B = 2^8$, except the user density μ is now reduced by a factor of 10.[5]

[5]If instead we considered the spectral efficiency $S = (\mu \times \text{user payload})$ versus E_b/N_0 trade-off, then the achievability region of these methods would be exactly the same as that obtained when the user payload is 8 bits and each user uses a single codebook of size $B = 2^8$. This is because the user payload of 80 bits (ten times that of 8 bits) cancels out the ten times reduction in the user density μ.

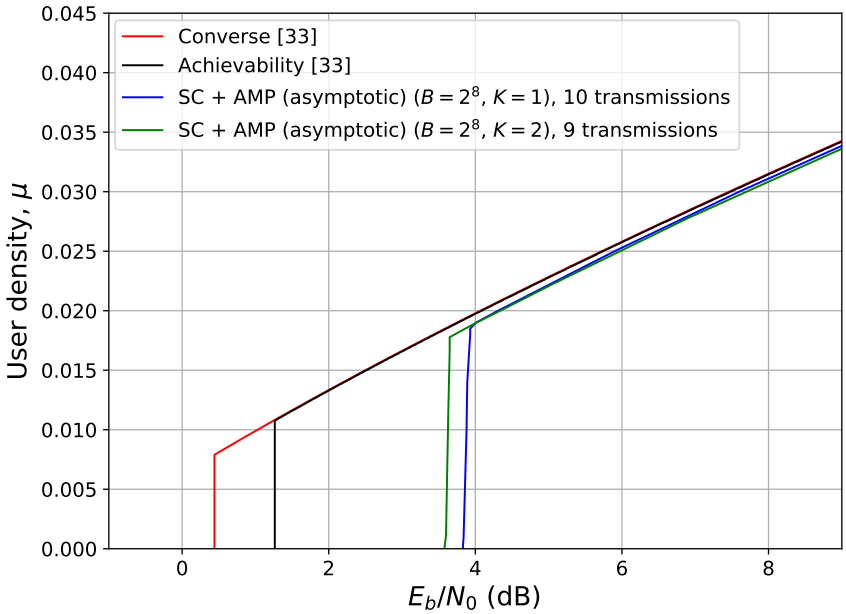


Figure 4.7: Red and black: converse and achievability bounds from [33] for massive multiple access when the per-user payload is 80 bits and the maximum tolerated UER is 10^{-3}. Green and blue: asymptotic achievable regions of coding methods based on spatially coupled Gaussian codebooks (of size $B = 2^8$) and AMP decoding for the same maximum tolerated UER. The $K = 2$ case (green) corresponds to encoding with spatially coupled Gaussian codebooks and binary modulation.

This (asymptotic) achievable region is shown in blue in Fig. 4.7, where we assumed that spatially coupled Gaussian codebooks (with $B = 2^8$) and AMP decoding are used, and the maximum tolerated UER is 10^{-3}. The red and black curves plot the converse and achievability bounds from [33] when the user payload is 80 bits and the maximum tolerated UER is also 10^{-3}. We see that one can still achieve near-optimal μ versus E_b/N_0 trade-offs using these methods at user densities above 0.02. However, at lower user densities, there is a noticeable gap between the achievability of these methods and the achievability of large ($B = 2^{80}$) i.i.d. Gaussian codebooks with ML decoding (black).

Although the two methods described above have the same effect on the achievability region, their effect on the computational complexity differs. Let L denote the number of users, B denote the size of the codebooks used in these methods ($B = 2^8$ in the above examples), and N denote either the number of transmissions in the first method, or the number of codebooks per user in the second method ($N = 10$ in the above examples). When DCT based codebooks are used, the complexity of the the first method scales as $O(NLB\log(LB))$ whereas the complexity of the second method scales as $O(NLB\log(NLB))$. Therefore, the first method is more efficient.

Using modulation to improve achievability

Let us consider again the example where the user payload is 80 bits. If in addition to encoding 8 bits using a size $B = 2^8$ codebook, we encode an extra 1 bit in the sign of the chosen codeword (as in Example 4.1.2), then each user's codeword would encode 9 bits, and 80 bits can be sent in

fewer than 9 transmissions. In comparison, to send 80 bits without modulation would require 10 transmissions of 8 bits each. The reduction in the number of transmissions reduces the complexity linearly, and the increase in AMP decoding complexity due to binary modulation is insignificant. Fig. 4.7 shows that using binary modulation (green) can increase the size of the achievable region.

4.4 Complex Gaussian channel and coding schemes

In this section we discuss how the coding methods described in this chapter and their analysis can be extended to *complex* Gaussian multiple access channels in the many-user setting. Recall that the coding methods described in this chapter can be represented using the random linear model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}+\boldsymbol{w}$, where $\boldsymbol{A}$ is an $n \times LB$ random design matrix which represents the codebooks of the L users, and the user messages are encoded in the message vector $\boldsymbol{x}$ which has L sections that are drawn i.i.d. from $p_{\boldsymbol{X}_{\text{sec}}}$ (a discrete distribution over length B vectors).

In the complex Gaussian channel, the noise vector $\boldsymbol{w}$ has i.i.d. circularly symmetric complex Gaussian entries with mean 0 and variance $N_0 = 2\sigma^2$, i.e., $w_i \sim \mathcal{CN}(0, N_0)$, with $\Re(w_i), \Im(w_i) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$, where $\Re(x)$ and $\Im(x)$ denote the real and imaginary part of a complex variable x.

Now the design matrix $\boldsymbol{A}$ and the message vector $\boldsymbol{x}$ can both have complex valued entries. The coding scheme that uses i.i.d. complex Gaussian codebooks corresponds to using a design matrix $\boldsymbol{A}$ with i.i.d. complex Gaussian entries $A_{ij} \sim \mathcal{CN}(0, \frac{1}{n})$ and a message vector $\boldsymbol{x}$ with sections that have a single non-zero entry equal to $\sqrt{E}$ (see also Example 4.1.1). Spatially coupled complex Gaussian design matrices $\boldsymbol{A}$ are constructed using base matrices $\boldsymbol{W} \in \mathbb{R}_+^{\mathsf{R}\times\mathsf{C}}$ in a similar way to their real-valued counterparts described in Section 4.1.1. They have independent complex Gaussian entries

$$A_{ij} \sim \mathcal{CN}\bigg(0, \frac{1}{n/\mathsf{R}} W_{\mathsf{r}(i)\mathsf{c}(j)}\bigg), \quad \text{for } i \in [n],\ j \in [LB]. \tag{4.82}$$

Recall that the L sections of $\boldsymbol{x}$ each correspond to a user's message, and the sections are drawn i.i.d. from $p_{\boldsymbol{X}_{\text{sec}}}$. Generalising the distribution p_2 that corresponded to random codebooks with binary modulation in Example 4.1.2, in this section we will often use the following example of $p_{\boldsymbol{X}_{\text{sec}}}$ which uses K-ary phase-shift keying (PSK) modulation.

Example 4.4.1 (Random codebooks with K-ary PSK modulation)**.** Let $p_{\boldsymbol{X}_{\text{sec}}}$ be the distribution over length B vectors that chooses uniformly at random one of its B entries to be non-zero, taking values in[6]

$$\Big\{ b_k \coloneqq \sqrt{E}\, e^{\mathrm{j}2\pi k/K} \Big\}_{k\in[K]} \tag{4.83}$$

[6]The PSK symbols are represented using $\{b_k\}_{k\in[K]}$ instead of $\{c_k\}_{k\in[K]}$ as done in Chapter 3 because in this chapter the vectors $\boldsymbol{c}_\ell$ for $\ell \in [L]$ correspond to the user codewords.

with equal probability, where $\mathrm{j} \coloneqq \sqrt{-1}$. Since each section of the message vector corresponds to B columns of the design matrix $\boldsymbol{A}$, in the coding scheme each user encodes $\log_2 B$ bits in the selection of one-of-B codewords, and an additional $\log_2 K$ bits in the selection of its phase. When the design matrix $\boldsymbol{A}$ has i.i.d. $\mathcal{CN}(0, \frac{1}{n})$ entries and $B = 1$, this coding scheme corresponds to random CDMA with K-ary PSK modulation. In the rest of the section, we denote the choice of $p_{\boldsymbol{X}_{\mathrm{sec}}}$ used in this example by p_K.

PSK modulation is considered instead of other modulation techniques because PSK symbols have equal magnitude. Unequal symbol magnitudes will counteract the effect of spatial coupling, and simulations show that errors are much more likely to occur in the sections where symbols of smaller magnitude are chosen. PSK modulation is considered for modulated complex sparse regression codes for the same reason (see Section 3.2).

Before we detail how the results in Theorems 4–7 can be extended to the complex Gaussian MAC in the many-user setting, we need to introduce the AMP decoder, state evolution, and potential function for the complex random linear coding setup.

AMP decoding The AMP algorithm for decoding the message vector $\boldsymbol{x}$ in the complex linear model and its corresponding state evolution only differ slightly from their real-valued counterparts described in Section 4.1.2.

The AMP decoder for spatially coupled and i.i.d. complex Gaussian $\boldsymbol{A}$ are the same as that given in (4.6) and (4.17), except the transpose operation is now the conjugate transpose operation. Furthermore, the denoising function η^t and the hard-decision MAP estimate of the message vector are defined using a standard complex Gaussian random vector $\boldsymbol{Z}$ instead of a standard real-valued one. That is, for index j in section $\ell \in [L]$, which we denote by $j \in \mathrm{sec}(\ell)$, with section ℓ in column block $\mathsf{c} \in [\mathsf{C}]$,

$$\eta_j^t(\boldsymbol{s}) = \mathbb{E}\left[(\boldsymbol{X}_{\mathrm{sec}})_j \mid \boldsymbol{X}_{\mathrm{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\, \boldsymbol{Z} = \boldsymbol{s}_\ell\right] \tag{4.84}$$

$$= \frac{\sum_{k=1}^K b_k \cdot \exp\left(\frac{\Re(\overline{s_j}\, b_k)}{\tau_{\mathsf{c}}^t}\right)}{\sum_{i \in \mathrm{sec}(\ell)} \sum_{k'=1}^K \exp\left(\frac{\Re(\overline{s_i}\, b_{k'})}{\tau_{\mathsf{c}}^t}\right)} \quad \text{if } \boldsymbol{X}_{\mathrm{sec}} \sim p_K, \tag{4.85}$$

where $\boldsymbol{X}_{\mathrm{sec}} \sim p_{\boldsymbol{X}_{\mathrm{sec}}}$ and $\boldsymbol{Z}$ is a standard complex Gaussian vector independent of $\boldsymbol{X}_{\mathrm{sec}}$. We use $\overline{z}$ to denote the complex conjugate of a complex number z. Recall that the ℓ-th section of a vector $\boldsymbol{s} \in \mathbb{R}^{LB}$ is denoted by $\boldsymbol{s}_\ell \in \mathbb{R}^B$ and p_K is described in Example 4.4.1. For section ℓ in column block $\mathsf{c} \in [\mathsf{C}]$, the ℓ-th section of the hard-decision MAP estimate after iteration $t+1$ is given by

$$\hat{\boldsymbol{x}}_\ell^{t+1} = \operatorname*{arg\,max}_{\boldsymbol{x}' \in \mathcal{S}} \mathbb{P}\Big(\boldsymbol{X}_{\mathrm{sec}} = \boldsymbol{x}' \mid \boldsymbol{X}_{\mathrm{sec}} + \sqrt{\tau_{\mathsf{c}}^t}\, \boldsymbol{Z} = \boldsymbol{s}_\ell^t\Big), \tag{4.86}$$

where $\mathcal{S}$ is the support of $p_{\boldsymbol{X}_{\mathrm{sec}}}$. When $\boldsymbol{X}_{\mathrm{sec}} \sim p_K$ with $K = 4$, i.e., the non-zero entries in $\boldsymbol{X}_{\mathrm{sec}}$ takes values in $\{\pm\sqrt{E}, \pm\mathrm{j}\sqrt{E}\}$ with equal probability, index $j \in \mathrm{sec}(\ell)$ of this hard-decision

estimate is given by

$$\hat{x}_j^{t+1} = \begin{cases} \operatorname{sign}(\Re(s_j)) \cdot \sqrt{E} & \text{if } |\Re(s_j)| > |\Im(s_j)| \text{ and} \\ & \quad |\Re(s_j)| > \max\{|\Re(s_i)|, |\Im(s_i)|\} \text{ for all } i \in \mathrm{sec}(\ell) \backslash j, \\ \operatorname{sign}(\Im(s_j)) \cdot \mathrm{j}\sqrt{E} & \text{if } |\Im(s_j)| > |\Re(s_j)| \text{ and} \\ & \quad |\Im(s_j)| > \max\{|\Re(s_i)|, |\Im(s_i)|\} \text{ for all } i \in \mathrm{sec}(\ell) \backslash j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.87}$$

The parameters $\{\tau_{\mathsf{c}}^t\}_{\mathsf{c} \in [\mathsf{C}]}$ are given by the state evolution recursion below.

State evolution The state evolution (SE) recursions of the AMP decoders for spatially coupled and i.i.d. complex Gaussian $\boldsymbol{A}$ are the same as that given in (4.7)–(4.8) and (4.18), except the noise variance term σ^2 is now N_0. Furthermore, the mmse function is defined using a standard complex Gaussian random vector $\boldsymbol{Z}$ instead of a standard real-valued one, i.e.,

$$\begin{aligned} &\mathsf{mmse}^{\mathrm{c}}(1/\tau) \\ &= \mathbb{E} \left\| \boldsymbol{X}_{\mathrm{sec}} - \mathbb{E}\Big[\boldsymbol{X}_{\mathrm{sec}} \mid \boldsymbol{X}_{\mathrm{sec}} + \sqrt{\tau} \boldsymbol{Z} \Big] \right\|^2 \qquad (4.88) \\ &= E - \mathbb{E} \left\{ \frac{\sum_{k=1}^K \sqrt{E}\, \Re(b_k) \cdot \exp\left(\frac{2\sqrt{E}\, \Re(b_k)}{\tau} + \frac{2\Re(Z_1 b_k)}{\sqrt{\tau}} \right)}{\sum_{k'=1}^K \left[\exp\left(\frac{2\sqrt{E}\, \Re(b_{k'})}{\tau} + \frac{2\Re(Z_1 b_{k'})}{\sqrt{\tau}} \right) + \sum_{j=2}^B \exp\left(\frac{2\Re(Z_j b_{k'})}{\sqrt{\tau}} \right) \right]} \right\} \quad \text{if } \boldsymbol{X}_{\mathrm{sec}} \sim p_K, \qquad (4.89) \end{aligned}$$

where $\boldsymbol{Z} = [Z_1, \ldots, Z_B]$ is a standard complex Gaussian random vector independent of $\boldsymbol{X}_{\mathrm{sec}}$, and the superscript "c" in $\mathsf{mmse}^{\mathrm{c}}$ denotes "complex". For example, the state evolution of the AMP decoder for i.i.d. complex Gaussian design matrices initialises $\psi^0 = \mathbb{E}\|\boldsymbol{X}_{\mathrm{sec}}^2\| = E$, and for $t \geq 0$ computes

$$\tau^t = N_0 + \mu \psi^t, \tag{4.90}$$

$$\psi^{t+1} = \mathsf{mmse}^{\mathrm{c}}(1/\tau^t). \tag{4.91}$$

Remark 4.4.1 (Real-valued $\boldsymbol{X}_{\mathrm{sec}}$)**.** When $p_{\boldsymbol{X}_{\mathrm{sec}}}$ is a distribution over real-valued random vectors, the effective noise variance in the definition of the $\mathsf{mmse}^{\mathrm{c}}$ function (4.88) is actually $\frac{\tau}{2}$ (instead of τ) since the MMSE (conditional expectation) estimator can ignore the noise along the imaginary axis. Therefore, if $\boldsymbol{X}_{\mathrm{sec}}$ is real, then

$$\mathsf{mmse}^{\mathrm{c}}(1/\tau) = \mathsf{mmse}(2/\tau), \tag{4.92}$$

where the mmse function is defined in (4.9), and the state evolution of the AMP decoder for

i.i.d. complex Gaussian design matrices (4.90)–(4.91) can be written as a single line recursion

$$\psi^{t+1} = \mathsf{mmse}\left(\frac{1}{\sigma^2 + \frac{\mu}{2}\psi^t}\right), \tag{4.93}$$

where we used $N_0 = 2\sigma^2$. Comparing (4.93) with the state evolution in the real-valued setting (4.18), we observe that when $\boldsymbol{X}_{\text{sec}}$ is real, the state evolution in the real and complex setting are exactly the same when the user density μ in the complex setting is twice the user density in the real setting. The same observation holds for the state evolution of the AMP decoder for spatially coupled complex Gaussian design matrices.

Potential function Recall that the results in Theorems 4 and 5 are given in terms of the minimisers or stationary points of the potential function defined in (4.20). The complex variants of these theorems are given in terms of the following potential function for user density $\mu = L/n > 0$ and channel noise variance $N_0 > 0$.

$$\mathcal{F}^{\text{c}}(\mu, N_0, \psi) = I(\boldsymbol{X}_{\text{sec}}; \boldsymbol{S}_\tau^{\text{c}}) + \frac{1}{\mu}\left[\ln\left(\frac{\tau}{N_0}\right) - \frac{\mu\psi}{\tau}\right], \tag{4.94}$$

where $\psi \in [0, E]$, $\tau = N_0 + \mu\psi$, and the mutual information is computed using the single-section complex Gaussian channel with variance τ:

$$\boldsymbol{S}_\tau^{\text{c}} = \boldsymbol{X}_{\text{sec}} + \sqrt{\tau}\boldsymbol{Z}, \tag{4.95}$$

where $\boldsymbol{X}_{\text{sec}} \sim p_{\boldsymbol{X}_{\text{sec}}}$ and $\boldsymbol{Z} \in \mathbb{C}^B$ is a standard complex Gaussian random vector independent of $\boldsymbol{X}_{\text{sec}}$. If $\boldsymbol{Z} = [Z_1, \ldots, Z_B]$ and $p_{\boldsymbol{X}_{\text{sec}}}$ is chosen to be the K-PSK modulated prior given in in Example 4.4.1, we have

$$\begin{aligned} I(\boldsymbol{X}_{\text{sec}}; \boldsymbol{S}_\tau^{\text{c}}) =& \frac{2E}{\tau} + \ln(BK) \\ &- \mathbb{E}\ln\left(\sum_{k=1}^{K}\left[\exp\left(\frac{2\sqrt{E}\Re(b_k)}{\tau} + \frac{2\Re(Z_1 b_k)}{\sqrt{\tau}}\right) + \sum_{j=2}^{B}\exp\left(\frac{2\Re(Z_j b_k)}{\sqrt{\tau}}\right)\right]\right). \end{aligned} \tag{4.96}$$

Remark 4.4.2 (Real-valued $\boldsymbol{X}_{\text{sec}}$)**.** When $p_{\boldsymbol{X}_{\text{sec}}}$ is a distribution over real-valued random vectors, the potential function $\mathcal{F}^{\text{c}}$ for the complex setting defined in (4.94) is equivalent to the potential function $\mathcal{F}$ for the real setting defined in (4.20) when the user density μ in the complex setting is twice the user density in the real setting, i.e.,

$$\mathcal{F}^{\text{c}}\left(\mu, N_0, \psi\right) = \mathcal{F}\left(\mu/2, \sigma^2, \psi\right). \tag{4.97}$$

This is because $N_0 = 2\sigma^2$ and in the calculation of the mutual information term $I(\boldsymbol{X}_{\text{sec}}; \boldsymbol{S}_\tau^{\text{c}})$, the effective noise variance of the single-section channel (4.95) is $\frac{\tau}{2}$ (instead of τ) due to $\boldsymbol{X}_{\text{sec}}$

being real.

Decoding error in single-section complex Gaussian channel Consider decoding $\boldsymbol{X}_{\text{sec}}$ in the complex Gaussian channel in (4.95). The MMSE decoder

$$\hat{\boldsymbol{x}}_{\text{sec}}^{\text{MMSE}}(\boldsymbol{S}_\tau^{\text{c}}) = \mathbb{E}[\boldsymbol{X}_{\text{sec}}|\boldsymbol{S}_\tau^{\text{c}}], \tag{4.98}$$

achieves the MMSE given by (4.88). The MAP decoder

$$\hat{\boldsymbol{x}}_{\text{sec}}^{\text{MAP}}(\boldsymbol{S}_\tau^{\text{c}}) = \arg\max_{\boldsymbol{x}'} \mathbb{P}(\boldsymbol{X}_{\text{sec}} = \boldsymbol{x}'|\boldsymbol{S}_\tau^{\text{c}}), \tag{4.99}$$

achieves the minimum probability of error, given by

$$P_e^{\text{c}}(\tau) = \mathbb{P}\big(\hat{\boldsymbol{x}}_{\text{sec}}^{\text{MAP}}(\boldsymbol{S}_\tau^{\text{c}}) \neq \boldsymbol{X}_{\text{sec}}\big) \tag{4.100}$$

$$= 1 - \int_{-\sqrt{\frac{2E}{\tau}}}^{\infty} \left[1 - 2Q\left(\sqrt{\frac{2E}{\tau}} + z\right)\right]^{2B-1} \phi(z)\,dz \quad \text{if } \boldsymbol{X}_{\text{sec}} \sim p_K \text{ and } K = 4, \tag{4.101}$$

where $\phi(\cdot)$ and $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-z^2/2}\,dz$ are the probability density function and upper tail probability of the standard (real-valued) Gaussian distribution, respectively.

Remark 4.4.3 (Real-valued $\boldsymbol{X}_{\text{sec}}$)**.** When $p_{\boldsymbol{X}_{\text{sec}}}$ is a distribution over real-valued random vectors, the minimum probability of decoding error in the single-section complex Gaussian channel (4.95) with noise variance τ is equal to the minimum probability of decoding error in the single-section real Gaussian channel (4.19) with noise variance $\frac{\tau}{2}$ since the optimal estimator can ignore the noise along the imaginary axis, i.e.,

$$P_e^{\text{c}}(\tau) = P_e(\tau/2), \tag{4.102}$$

where $P_e(\cdot)$ is defined in (4.25).

Notice that the probability of error function for 4-PSK modulated sections in the complex noise setting given in (4.101) is the same as that for binary modulated (2-PSK) sections in the real noise setting given in (4.26), except that τ in (4.26) changed to $\frac{\tau}{2}$ (due to moving from real to complex), and B in (4.26) changed to $2B$.

4.4.1 Theoretical results

Asymptotic UER achieved by AMP decoding

The complex variants of Theorems 4 and 5 describe the asymptotic UER achieved by AMP decoding when the design matrix $\boldsymbol{A}$ is either i.i.d. complex Gaussian, or spatially coupled complex Gaussian constructed using an (ω, Λ, ρ) base matrix, and the channel is complex Gaussian. We

do not explicitly state Theorems 4 and 5 for the complex setting, but we expect these results to be the same as Theorems 4 and 5, except that they are given in terms of the minimiser(s) and the largest stationary point of the potential function $\mathcal{F}^{\mathrm{c}}$ defined in (4.94), and they use the probability of error function P_e^{c} defined in (4.100).

Remark 4.4.4 (Real-valued $\boldsymbol{X}_{\mathrm{sec}}$)**.** When $p_{\boldsymbol{X}_{\mathrm{sec}}}$ is a distribution over real-valued random vectors, the asymptotic UER achieved by AMP decoding in the complex setting (with either i.i.d. or spatially coupled design matrices) at user density μ is the same as that achieved in the real setting at user density $\mu/2$. (See Remarks 4.4.1, 4.4.2 and 4.4.3, and note that $\frac{\tau}{2} = \sigma^2 + \frac{\mu}{2}\psi$ in the complex setting.)

To prove the complex variants of Theorems 4 and 5, one needs to prove that the performance of the AMP decoder is accurately tracked by the state evolution recursion for general complex discrete priors $p_{\boldsymbol{X}_{\mathrm{sec}}}$ and both i.i.d. and spatially coupled complex Gaussian design matrices. (See (4.36) and (4.54) in the proofs of Theorems 4 and 5.) Such a concentration result was proved in [160] for the special case where the design matrix has i.i.d. complex Gaussian entries and the message vector has i.i.d. entries ($B = 1$). For the complex design matrices and message vectors under consideration, the proof is essentially the same as that in [1,22,57], which consider real-valued design matrices and the specific $p_{\boldsymbol{X}_{\mathrm{sec}}}$ distribution given in Example 4.1.1. Apart from this AMP result for the complex setting, the proofs of the complex variants of Theorems 4 and 5 are nearly identical to those of Theorems 4 and 5 due to the similarity of the state evolution recursions and potential functions in the real and complex setting.

Large user payloads

The complex variants of Theorems 6 and 7 provide bounds on the asymptotic UER achieved by AMP decoding at large user payloads when complex Gaussian codebooks (without modulation) are used, i.e., when the sections of the message vector have a single non-zero entry equal to $\sqrt{E}$.

Since the asymptotic UER achieved by AMP decoding in the complex setting is the same as that achieved in the real setting at half the user density when the message vector is real (Remark 4.4.4), the complex variants of Theorems 6 and 7 are exactly the same as Theorems 6 and 7 except with the user density μ and spectral efficiency $S = \mu \log_2 B$ replaced with $\frac{\mu}{2}$ and $\frac{S}{2}$. The proofs are the same as those for Theorems 6 and 7, except one needs to prove that the performance of the AMP decoder is accurately tracked by state evolution in the complex setting (as discussed earlier).

The complex variants of Theorems 6 and 7 consider complex Gaussian codebooks without modulation. We also expect similar results to hold for complex Gaussian codebooks with K-ary PSK modulation (Example 4.4.1) when K is a power of 2, albeit with the bounds on the asymptotic UER being a function of K. In the proofs, the upper bound in Lemma 4.3.1 will need to be modified according to Lemma 3.4.1, and the bounds on the (normalised) state evolution

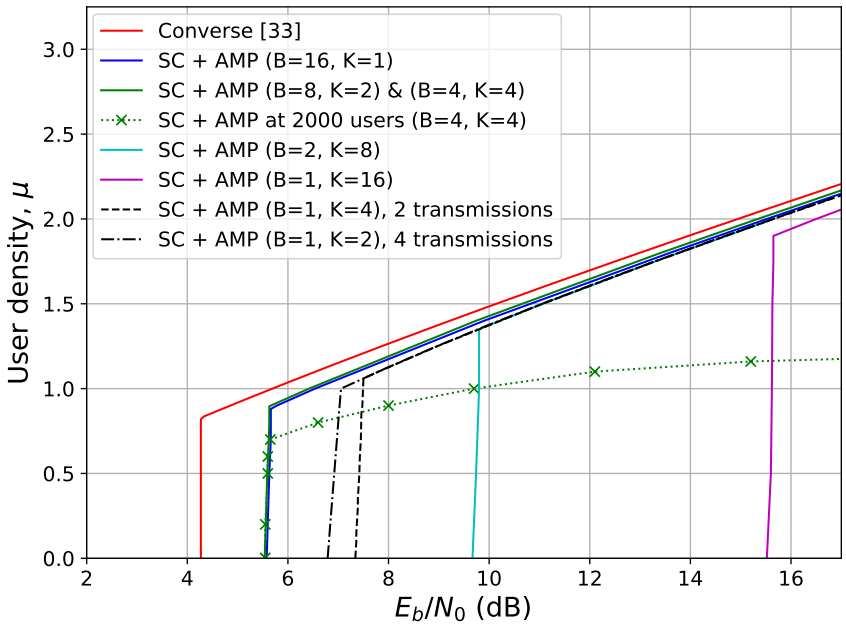


Figure 4.8: Achievable regions for complex massive multiple access when the per-user payload is 4 bits and the maximum tolerated UER is 10^{-3}. Users encode their information using random codebooks of size B (defined by the design matrix) and K-ary PSK modulation.

parameter $\frac{\psi^{t+1}}{E}$ (e.g. in (4.67) and (4.79)) modified according to Propositions 3.4.1 and 3.4.2. We omit the details here.

4.4.2 Numerical results

In Fig. 4.8, we plot the asymptotic achievable region of the coding scheme based on spatially coupled complex Gaussian design matrices and AMP decoding, when the user payload is 4 bits and the maximum tolerated UER is 10^{-3}.

One benefit of the complex setting is that complex modulation techniques can be used to encode information in the message vector. In these simulations the users encode their information in K-PSK modulated message vector sections of length B (Example 4.4.1); hence, $\log_2(BK) = 4$ bits. In Fig. 4.8 we consider the following combinations of B and K:

- $(B = 16,\ K = 1)$, which corresponds to unmodulated sections (blue),
- $(B = 8,\ K = 2)$, which corresponds to binary modulated sections (green),
- $(B = 4,\ K = 4)$, which corresponds to 4-PSK modulated sections (green),
- $(B = 2,\ K = 8)$, which corresponds to 8-PSK modulated sections (cyan), and
- $(B = 1,\ K = 16)$, which corresponds to random CDMA with 16-PSK modulation (magenta).

Table 4.2: Optimised coupling width values used in Fig. 4.8.

μ	0.70	0.80	0.90	1.00	1.10	1.16	1.18
ω	3	3	3	4	5	6	6

We compare our results with the converse bound in [33] (red), doubling the user density since we are considering complex instead of real channels. We also compare our results to random CDMA with either binary (dashed and dotted black) or 4-PSK (dashed black) modulation, and MAP decoding. The random CDMA with binary modulation (1 bit) and 4-PSK modulation (2 bits) are used 4 times and 2 times, respectively, to complete the 4 bit user payload.

From Fig. 4.8 we see that the asymptotic achievable region for the ($B = 8$, $K = 2$) and ($B = 4$, $K = 4$) encoding scheme are exactly the same. This is because the potential function in (4.94) for K-PSK modulated sections has the same expression for ($B = B'$, $K = 2$) and ($B = B'/2$, $K = 4$), and from Remark 4.4.3 we know that $P_e^{\mathrm{c}}(\cdot)$ also has the same expression for ($B = B'$, $K = 2$) and ($B = B'/2$, $K = 4$).

Comparing the different complex encoding schemes, we see that the ($B = 8$, $K = 2$) and ($B = 4$, $K = 4$) encoding schemes have the largest asymptotic achievable region, closely followed by ($B = 16$, $K = 1$), and then by ($B = 2$, $K = 8$) and ($B = 1$, $K = 16$). There is a large gap between the ($B = 16$, $K = 1$) and ($B = 2$, $K = 8$) encoding scheme, and an even larger gap between the ($B = 2$, $K = 8$) and ($B = 1$, $K = 16$) encoding scheme. The asymptotic achievable regions of random CDMA with binary and 4-PSK modulation come in between that of the ($B = 16$, $K = 1$) and ($B = 2$, $K = 8$) encoding schemes, with binary modulation having the larger achievable region of the two.

We also show the simulated performance of the ($B = 4$, $K = 4$) spatially coupled coding scheme at $L = 2000$ users (green dotted line with crosses). For a list of user densities μ, the crosses show the minimum E_b/N_0 at which the coding scheme achieves an average UER less than 10^{-3} (averaged over many independent trials). The spatially coupled complex design matrix was constructed using (ω, $\Lambda = 20$, $\rho = 0$) base matrices, with the coupling width ω optimised for each user density μ (see Table 4.2). Furthermore, we use discrete Fourier transform (DFT) based design matrices and the fast Fourier transform (FFT) is used to reduce complexity and memory requirements (see Section 2.5.1 for implementation details).

Similar to Fig. 4.4, we see that the achievable region of the finite user simulations matches the asymptotic achievable region (solid green) at low user densities. However, above a certain user density threshold, roughly $\mu = 0.70$ in Fig. 4.8, the gap to the asymptotic curve widens as the user density increases due to the relatively small values of base matrix parameters. Moreover, from Table 4.2 we observe that the optimal coupling width ω increases with the user density above the user density threshold.

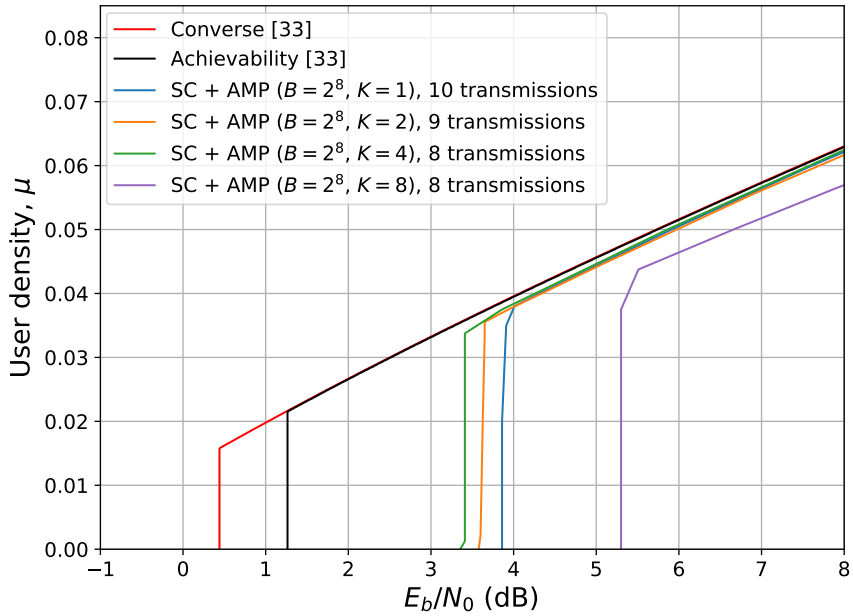


Figure 4.9: Red and black: converse and achievability bounds from [33] for massive multiple access when the per-user payload is 80 bits and the maximum tolerated UER is 10^{-3}. Blue, orange, green and purple: asymptotic achievable regions of coding methods based on spatially coupled complex Gaussian codebooks (of size $B = 2^8$) with K-ary PSK modulation and AMP decoding for the same maximum tolerated UER.

Large user payloads

Just as introducing binary modulation to the encoding scheme can improve the achievable region and reduce computational complexity at large user payloads (end of Section 4.3.4), introducing K-ary PSK modulation can have the same effect in the complex setting.

Fig. 4.9 considers the same setting as in Fig. 4.7, except with a complex Gaussian channel and K-ary PSK modulation. The user payload is 80 bits and the maximum tolerated UER is 10^{-3}. Since it is not feasible to implement large codebooks of size $B = 2^{80}$, we use smaller codebooks of size $B = 2^8$ with varying levels of modulation multiple times, with the modulation factor being K. Each transmission encodes $\log_2(BK)$ bits. For example, the $(B = 2^8, K = 1)$ scheme encodes 8 bits/transmission and requires 10 transmissions to complete the payload of 80 bits, and the $(B = 2^8, K = 4)$ scheme encodes 10 bits/transmission and requires 8 transmissions.

Fig. 4.9 shows that using 2-PSK (orange) and 4-PSK (green) modulation increases the size of the achievable region; however, using 8-PSK (purple) decreases the size of the achievable region. Moreover, as the modulation factor K increases, the number of transmissions required (hence the computational complexity) is reduced. The increase in AMP decoding complexity due to increased modulation is insignificant when $K \ll \log(LB)$ (see Section 3.5.1).

Chapter 5

Conclusions

In this thesis we investigated the sparse regression code (SPARC) framework with efficient approximate message passing (AMP) decoding for capacity-achieving communications over the single-user additive white Gaussian noise (AWGN) channel and the Gaussian multiple access channel (MAC) in the many-user setting.

In Chapter 2, we introduced spatially coupled (SC) SPARCs constructed via base matrices, where the design matrix used to define the code has a block-diagonal structure. We showed that base matrices provide a unified framework for designing power allocated and spatially coupled SPARCs. We described an AMP decoder for SC-SPARCs and analysed its decoding progression in the regime where the sections of the message vector are large. The analysis explained the "wave-like" decoding progression in SC-SPARCs (Fig. 2.2). We showed that for any rate less than the capacity, and sufficiently large coupling width ω, coupling length Λ and section size M, the probability of excess section error rate of the AMP decoder after T iterations decays exponentially in the code length n (Theorem 1), where T is inversely proportional to the rate gap to capacity.

The finite code length simulation results in Section 2.5 showed that SC-SPARCs with AMP decoding can achieve lower error rates than power allocated SPARCs and standard coded modulation schemes based on LDPC codes and QAM modulation. Motivated by the wave-like decoding progression of SC-SPARCs, we introduced a sliding window AMP decoder where a decoding window moves (unidirectionally) across the message vector, updating only the estimates of the sections within the decoding window at each window position. The latency and per-iteration decoding complexity of this decoder is independent of the code length.

In Chapter 3, we proposed *modulated* SPARCs, a generalisation of SPARCs for the complex AWGN channel, where the user message is encoded in both the locations and the values of the non-zero entries of the message vector. This generalisation introduces more flexibility in the SPARC code design, allowing us to reduce decoding complexity without affecting error performance (at a given rate). We considered the setting where the values of the non-zero entries of the message vector were chosen from a K-ary phase shift keying (PSK) constellation

to ensure that the modulation scheme does not counteract the effect of power allocation or spatial coupling.

We analysed the performance of an AMP decoder by obtaining analytical bounds on a key state evolution parameter which predicts its MSE in each iteration (Proposition 3.4.1). These bounds showed that in the large system limit ($L, M, n \to \infty$ with fixed $R = L\log(MK)/n$ and K), the per-iteration MSE of the AMP decoder for K-PSK modulated SPARCs does not depend on the modulation factor K. Therefore, the asymptotic analysis of K-PSK modulated SPARCs is equivalent to that of unmodulated ($K = 1$) SPARCs. This equivalence together with existing state evolution analysis for unmodulated SPARCs was used to show that in the large system limit, for any rate less than the capacity, K-PSK modulated SPARCs with suitable power allocation or spatial coupling and AMP decoding achieves zero section error rate in a finite number of iterations that depends on the rate gap to capacity.

In Chapter 4 we considered Gaussian MACs in the asymptotic setting where the number of users L and the code length n both tend to infinity with the user density $\mu = L/n$ held constant, and where the user payload, user energy and user error rate (the fraction of user messages decoded in error) are fixed. We analysed coding schemes based on the random linear models (e.g., SPARCs) and efficient AMP decoding, and derived the exact asymptotic achievable trade-off between the signal-to-noise ratio E_b/N_0 and the user density μ, for a fixed target user error rate and user payload. The SPARC framework fit naturally to this problem setting as each section of the message vector represents a user's message.

We found that the asymptotic achievable region of a coding scheme based on *spatially coupled* Gaussian matrices and AMP decoding exceeds that obtained using the achievability bound in [33] and nearly matches the converse bound for a large range of user densities. To the best of our knowledge, this is the first efficient coding scheme to do so in this MAC regime. The spatially coupled scheme can be interpreted as generalised time-sharing: the coupling structure specifies which users are active during each channel use. We also analysed the performance of these coding schemes as the user payload grows large and extended our results to the complex Gaussian MAC. We showed that using small random codebooks multiple times to transmit large user payloads is near-optimal at large user densities, and adding modulation (e.g., K-PSK modulation) to the encoding scheme can increase the size of the asymptotic achievable region in such settings.

5.1 Future directions

Base matrix designs In this thesis we introduced SPARCs where the design matrix used to define the code is constructed using a base matrix. We explained how power allocated and spatially coupled SPARCs can be represented using simple base matrices, e.g., the (ω, Λ, ρ) base matrix. These simple base matrix designs were used to prove that power allocated and spatially

coupled SPARCs with K-PSK modulation are asymptotically capacity achieving; however, they are not optimised for practical use.

We would like to develop base matrix designs that give better finite code length error performance. One idea is to use both power allocation and spatial coupling in the design (see Remark 2.1.2). Preliminary simulation results show that it is possible for such a combined design to achieve lower error rates at finite lengths than SPARCs with either power allocation or spatial coupling. However, this design is harder to optimise due to the increased number of design parameters.

Moreover, the base matrix framework and the base matrix designs for SPARCs can be applied to the coding schemes introduced in Chapter 4 for the Gaussian MAC in the many-user setting, and may also inspire better designs in other random linear (or generalised linear) estimation problems. For example, in the noiseless compressed sensing problem, we would like to discover a simple base matrix design that can demonstrate *in practice* that the theoretically optimal undersampling ratio can be obtained with spatial coupling and AMP algorithms. (The base matrix design in [139] is sufficient for the proof of this result, but does not obtain near-optimal undersampling ratios in practice.) We expect that effective spatially coupled base matrix designs would differ depending on whether the compressed sensing signal entries are discrete or continuous.

Base matrix designs for sliding window AMP decoding The unidirectional sliding window AMP decoder introduced in Section 2.6 for SC-SPARCs does not fully utilise the "seeds" at the two ends (corners) of the $(\omega, \Lambda, \rho = 0)$ base matrix. We would like to explore other base matrix designs that are tailored to the unidirectional window decoder. We expect such designs to have a larger seed region at the top-left corner of the base matrix, and a smaller (if not none at all) seed region at the bottom-right corner of the base matrix.

Implementation of fast transforms for spatially coupled design matrices In Section 2.5.1, we discussed how we encode and decode with Hadamard and Fourier based spatially coupled design matrices using their corresponding fast transforms (FWHT and FFT). The implementation was simple: each block of the spatially coupled design matrix corresponded to one fast transform operation. It is possible to consider other implementations that may lead to advantages in design simplicity and computational complexity, such as doing one fast transform per ω (the coupling width) blocks (either blocks within the same row-block or column-block).

Closing the converse and achievability gap in many-user MACs There exists a gap between the converse and achievability regions of many-user Gaussian MACs at low user payloads (see Chapter 4, e.g., Fig. 4.4). We showed (via numerical simulation results) that using modulation in addition to random codebooks can narrow the gap at very small user payloads

(2 bits), but has no effect on the achievability region at 8 bits per user. We would like to know whether there are coding schemes whose achievability matches the converse at the low user densities, or whether the converse bound is not tight.

Evaluating modern multiple access techniques for many-user MACs In Chapter 4 we considered Gaussian MACs in the many-user setting and showed that the asymptotic achievability of a coding scheme based on spatially coupled Gaussian matrices and AMP decoding is near-optimal at large user densities. We would like to understand how the achievability of modern multiple access techniques such as sparse code multiple access and non-orthogonal multiple access compare to the achievability of our proposed coding scheme (both asymptotically and at finite number of users).

Inspiring new multiple access techniques The multiple access technique we introduced in Chapter 4 was based on spatially coupled Gaussian random codebooks and PSK modulation. We would be interested to see if this approach inspires deterministic multiple access techniques with better performance. Recall from Section 4.1.1 that spatial coupling can be viewed as block-wise time (or frequency) division with overlap, and random codebooks with modulation can be seen as a generalisation of random CDMA (Example 4.4.1).

In the linear coding framework of (4.2), one can use a design matrix with specific structural properties (rather than the i.i.d. Gaussian design). Examples include matrices where the user codewords are orthogonal (or approximately orthogonal) and matrices that facilitate certain low complexity encoders or decoders. Our simulations in Section 4.4.2 already use DFT based spatially coupled design matrices and the FFT algorithm for low complexity encoding and decoding; perhaps there are ways to optimise this design for practical implementation.

Furthermore, instead of having users encode their message in the location and value of the single non-zero entry in a section of the message vector, different encoding schemes ($p_{\boldsymbol{X}_{\text{sec}}}$ distributions) can be considered.

Unsourced random access SPARCs have been considered for unsourced random access in [38–40]. It would be interesting to explore how the spatial coupling and modulation techniques discussed in this thesis can be applied to that problem setting.

Bibliography

[1] C. Rush, K. Hsieh, and R. Venkataramanan, "Capacity-achieving spatially coupled sparse superposition codes with AMP decoding," to appear in IEEE Trans. Inf. Theory. [Online]. Available: https://arxiv.org/abs/2002.07844

[2] K. Hsieh, C. Rush, and R. Venkataramanan, "Spatially coupled sparse regression codes: Design and state evolution analysis," in *Proc. IEEE Int. Symp. Inf. Theory*, June 2018, pp. 1016–1020.

[3] C. Rush, K. Hsieh, and R. Venkataramanan, "Capacity-achieving sparse regression codes via spatial coupling," in *Proc. IEEE Inf. Theory Workshop*, 2018.

[4] C. Rush, K. Hsieh, and R. Venkataramanan, "Spatially coupled sparse regression codes with sliding window AMP decoding," in *Proc. IEEE Inf. Theory Workshop*, 2019.

[5] K. Hsieh and R. Venkataramanan, "Modulated sparse superposition codes for the complex AWGN channel," to appear in IEEE Trans. Inf. Theory. [Online]. Available: https://arxiv.org/abs/2004.09549

[6] K. Hsieh and R. Venkataramanan, "Modulated sparse regression codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 1432–1437.

[7] K. Hsieh, C. Rush, and R. Venkataramanan, "Near-optimal coding for massive multiple access," to appear in Proc. IEEE Int. Symp. Inf. Theory, 2021. [Online]. Available: https://arxiv.org/abs/2102.04730

[8] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948.

[9] T. M. Cover and J. A. Thomas, *Elements of information theory.* John Wiley & Sons, 2006.

[10] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct 1996.

[11] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[12] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar 1999.

[13] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[14] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb 2011.

[15] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec 2013.

[16] D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, June 2007.

[17] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. 28, no. 1, pp. 55–67, Jan 1982.

[18] G. D. Forney and G. Ungerboeck, "Modulation and coding for linear gaussian channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2384–2415, Oct 1998.

[19] A. Guillén i Fàbregas, A. Martinez, and G. Caire, "Bit-interleaved coded modulation," *Found. Trends Commun. Inf. Theory*, vol. 5, no. 1–2, pp. 1–153, 2008.

[20] A. Joseph and A. R. Barron, "Least squares superposition codes of moderate dictionary size are reliable at rates up to capacity," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2541–2557, May 2012.

[21] A. Joseph and A. R. Barron, "Fast sparse superposition codes have near exponential error probability for $R < \mathcal{C}$," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 919–942, Feb. 2014.

[22] C. Rush, A. Greig, and R. Venkataramanan, "Capacity-achieving sparse superposition codes via approximate message passing decoding," *IEEE Trans. Inf. Theory*, vol. 63, no. 3, pp. 1476–1500, Mar. 2017.

[23] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[24] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 371–377, May 1977.

[25] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: theoretical concepts and practical design rules," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1361–1391, Jul 1999.

[26] R. Zamir, B. Nazer, Y. Kochman, and I. Bistritz, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation and Multiuser Information Theory.* Cambridge University Press, 2014.

[27] A. El Gamal and Y.-H. Kim, *Network information theory.* Cambridge university press, 2011.

[28] X. Chen, T. Chen, and D. Guo, "Capacity of Gaussian many-access channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3516–3539, June 2017.

[29] Y. Wu, X. Gao, S. Zhou, W. Yang, Y. Polyanskiy, and G. Caire, "Massive access for future wireless communication systems," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 148–156, 2020.

[30] J. Ravi and T. Koch, "Capacity per unit-energy of Gaussian many-access channels," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 2763–2767.

[31] ——, "On the per-user probability of error in Gaussian many-access channels," in *Proc. Int. Zurich Seminar Inf. Commun.*, 2020, pp. 139 – 143.

[32] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 2523–2527.

[33] I. Zadik, Y. Polyanskiy, and C. Thrampoulidis, "Improved bounds on Gaussian MAC and sparse regression via Gaussian inequalities," in *Proc. IEEE Int. Symp. Inf. Theory*, July 2019, pp. 430–434.

[34] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[35] S. S. Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy, "Energy efficient random access for the quasi-static fading MAC," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 2768–2772.

[36] S. S. Kowshik and Y. Polyanskiy, "Fundamental limits of many-user MAC with finite payloads and fading," 2019. [Online]. Available: https://arxiv.org/abs/1901.06732

[37] J. Ravi and T. Koch, "Capacity per unit-energy of Gaussian random many-access channels," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 3025–3030.

[38] V. K. Amalladinne, A. K. Pradhan, C. Rush, J.-F. Chamberland, and K. R. Narayanan, "On approximate message passing for unsourced access with coded compressed sensing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 2995–3000.

[39] A. Fengler, P. Jung, and G. Caire, "SPARCs and AMP for unsourced random access," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 2843–2847.

[40] ——, "Unsourced multiuser sparse regression codes achieve the symmetric MAC capacity," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 3001–3006.

[41] J. Barbier, M. Dia, and N. Macris, "Threshold saturation of spatially coupled sparse superposition codes for all memoryless channels," *Proc. IEEE Inf. Theory Workshop*, 2016.

[42] J. Barbier, M. Dia, and N. Macris, "Universal sparse superposition codes with spatial coupling and GAMP decoding," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5618–5642, Sept. 2019.

[43] R. Venkataramanan, A. Joseph, and S. Tatikonda, "Lossy compression via sparse linear regression: Performance under minimum-distance encoding," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3254–3264, June 2014.

[44] R. Venkataramanan and S. Tatikonda, "Sparse regression codes for multi-terminal source and channel coding," in *Proc. 50th Annual Allerton Conf. Commun., Control, Comput.*, Oct 2012, pp. 1966–1974.

[45] R. Venkataramanan, S. Tatikonda, and A. Barron, "Sparse regression codes," *Found. Trends Commun. Inf. Theory*, vol. 15, no. 1-2, pp. 1–195, 2019.

[46] S. Cho and A. R. Barron, "Approximate iterative Bayes optimal estimates for high-rate sparse superposition codes," in *Sixth Workshop on Inf. Theoretic Methods in Sci. and Eng.*, 2013, pp. 35–42.

[47] A. Greig and R. Venkataramanan, "Techniques for improving the finite length performance of sparse superposition codes," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 905–917, Mar. 2018.

[48] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[49] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[50] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[51] Y. Takeishi, M. Kawakita, and J. Takeuchi, "Least squares superposition codes with Bernoulli dictionary are still reliable at rates up to capacity," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2737–2750, May 2014.

[52] Y. Takeishi and J. Takeuchi, "An improved upper bound on block error probability of least squares superposition codes with unbiased Bernoulli dictionary," in *Proc. IEEE Int. Symp. Inf. Theory*, July 2016, pp. 1168–1172.

[53] A. R. Barron and S. Cho, "High-rate sparse superposition codes with iteratively optimal estimates," *Proc. IEEE Int. Symp. Inf. Theory*, 2012.

[54] J. Barbier and F. Krzakala, "Replica analysis and approximate message passing decoder for superposition codes," *Proc. IEEE Int. Symp. Inf. Theory*, 2014.

[55] J. Barbier, C. Schülke, and F. Krzakala, "Approximate message-passing with spatially coupled structured operators, with applications to compressed sensing and sparse superposition codes," *J. Stat. Mech.: Theory Exp.*, vol. 2015, no. 5, p. P05013, 2015.

[56] J. Barbier and F. Krzakala, "Approximate message-passing decoder and capacity achieving sparse superposition codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 4894–4927, Aug. 2017.

[57] C. Rush and R. Venkataramanan, "The error probability of sparse superposition codes with approximate message passing decoding," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3278–3303, May 2019.

[58] A. Joseph, "Achieving information-theoretic limits with high-dimensional regression," Ph.D. dissertation, Yale University, 2012.

[59] J. L. Shanks, "Computation of the fast Walsh-Fourier transform," *IEEE Transactions on Computers*, vol. C-18, no. 5, pp. 457–459, May 1969.

[60] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[61] C. Condo and W. J. Gross, "Sparse superposition codes: A practical approach," *Proc. IEEE Int. Workshop on Signal Processing Systems*, 2015.

[62] ——, "Implementation of sparse superposition codes," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2421–2427, 2017.

[63] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[64] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory*, July 2011, pp. 2168–2172.

[65] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová, "Statistical-physics-based reconstruction in compressed sensing," *Phys. Rev. X*, vol. 2, p. 021005, May 2012.

[66] A. Montanari and R. Venkataramanan, "Estimation of low-rank matrices via approximate message passing," *The Annals of Statistics*, vol. 49, no. 1, pp. 321 – 345, 2021.

[67] J. Boutros and G. Caire, "Iterative multiuser joint decoding: unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, 2002.

[68] T. Tanaka and M. Okada, "Approximate belief propagation, density evolution, and statistical neurodynamics for CDMA multiuser detection," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 700–706, 2005.

[69] A. Montanari and D. Tse, "Analysis of belief propagation for non-linear problems: The example of CDMA (or: How to prove Tanaka's formula)," in *Proc. IEEE Inf. Theory Workshop*, 2006, pp. 160–164.

[70] D. Guo and C. Wang, "Asymptotic mean-square optimality of belief propagation for sparse linear systems," in *Proc. IEEE Inf. Theory Workshop*, 2006, pp. 194–198.

[71] D. L. Donoho, A. Maleki, and A. Montanari, "The noise-sensitivity phase transition in compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6920–6941, 2011.

[72] M. Bayati and A. Montanari, "The LASSO risk for Gaussian matrices," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 1997–2017, Apr. 2012.

[73] G. Reeves and M. Gastpar, "The sampling rate-distortion tradeoff for sparsity pattern recovery in compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3065–3092, 2012.

[74] G. Reeves and H. D. Pfister, "The replica-symmetric prediction for random linear estimation with Gaussian matrices is exact," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2252–2283, 2019.

[75] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Proc. IEEE Inf. Theory Workshop*, 2010.

[76] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[77] A. Montanari, *Graphical models concepts in compressed sensing.* Cambridge University Press, 2012, pp. 394–438.

[78] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices," *J. Stat. Mech.: Theory Exp.*, vol. 2012, no. 08, p. P08009, 2012.

[79] T. Richardson and R. Urbanke, *Modern Coding Theory.* New York, NY, USA: Cambridge University Press, 2008.

[80] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[81] ——, "Minimax risk over ℓ_p-balls for ℓ_q-error," *Probab. Theory Relat. Fields*, vol. 99, no. 2, pp. 277–303, 1994.

[82] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[83] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of Maxwell saturation for coupled scalar recursions," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6943–6965, 2014.

[84] J. Barbier, N. Macris, M. Dia, and F. Krzakala, "Mutual information and optimality of approximate message-passing in random linear estimation," *IEEE Trans. Inf. Theory*, vol. 66, no. 7, pp. 4270–4303, 2020.

[85] D. Guo, S. Shamai, and S. Verdú, "Mutual information and minimum mean-square error in Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1261–1282, 2005.

[86] J. S. Yedidia, W. T. Freeman, and Y. Weiss, *Understanding Belief Propagation and Its Generalizations.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 239–269.

[87] ——, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[88] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[89] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, 2012.

[90] J. Tan, Y. Ma, and D. Baron, "Compressive imaging via approximate message passing with image denoising," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2085–2092, 2015.

[91] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "BM3D-AMP: A new image recovery algorithm based on BM3D denoising," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 3116–3120.

[92] ——, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, 2016.

[93] Y. Ma, C. Rush, and D. Baron, "Analysis of approximate message passing with a class of non-separable denoisers," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 231–235.

[94] ——, "Analysis of approximate message passing with non-separable denoisers and Markov random field priors," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, pp. 7367–7389, 2019.

[95] R. Berthier, A. Montanari, and P.-M. Nguyen, "State evolution for approximate message passing with non-separable functions," *Information and Inference: A Journal of the IMA*, vol. 9, no. 1, pp. 33–79, 01 2019.

[96] J. Vila and P. Schniter, "Expectation-maximization Bernoulli-Gaussian approximate message passing," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, 2011, pp. 799–803.

[97] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, 2013.

[98] C. Guo and M. E. Davies, "Near optimal compressed sensing without priors: Parametric SURE approximate message passing," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2130–2141, 2015.

[99] Y. Ma, J. Zhu, and D. Baron, "Approximate message passing algorithm with universal denoising and Gaussian mixture learning," *IEEE Trans. Signal Process.*, vol. 64, no. 21, pp. 5611–5622, 2016.

[100] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Information and Inference: A Journal of the IMA*, vol. 2, no. 2, pp. 115–144, 12 2013.

[101] J. Barbier, F. Krzakala, N. Macris, L. Miolane, and L. Zdeborová, "Optimal errors and phase transitions in high-dimensional generalized linear models," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 116, no. 12, pp. 5451–5460, 2019.

[102] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1043–1055, 2015.

[103] D. Donoho and A. Montanari, "High dimensional robust M-estimation: asymptotic variance via approximate message passing," *Probab. Theory Relat. Fields*, vol. 166, no. 3, pp. 935–969, 2016.

[104] P. Sur, Y. Chen, and E. J. Candès, "The likelihood ratio test in high-dimensional logistic regression is asymptotically a rescaled Chi-square," *Probab. Theory Relat. Fields*, vol. 175, no. 1, pp. 487–558, 2019.

[105] P. Sur and E. J. Candès, "A modern maximum-likelihood theory for high-dimensional logistic regression," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 116, no. 29, pp. 14 516–14 525, 2019.

[106] Z. Bu, J. M. Klusowski, C. Rush, and W. J. Su, "Algorithmic analysis and statistical estimation of SLOPE via approximate message passing," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 506–537, 2021.

[107] M. Bayati, M. Lelarge, and A. Montanari, "Universality in polytope phase transitions and message passing algorithms," *The Annals of Applied Probability*, vol. 25, no. 2, pp. 753 – 822, 2015.

[108] T. P. Minka, *Expectation Propagation for Approximate Bayesian Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.

[109] M. Opper and O. Winther, "Expectation consistent approximate inference," *Journal of Machine Learning Research*, vol. 6, no. 73, pp. 2177–2204, 2005.

[110] B. Çakmak, O. Winther, and B. H. Fleury, "S-AMP: Approximate message passing for general matrix ensembles," in *Proc. IEEE Inf. Theory Workshop*, 2014, pp. 192–196.

[111] A. Fletcher, M. Sahraee-Ardakan, S. Rangan, and P. Schniter, "Expectation consistent approximate inference: Generalizations and convergence," in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 190–194.

[112] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[113] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6664–6684, 2019.

[114] S. Rangan, P. Schniter, A. K. Fletcher, and S. Sarkar, "On the convergence of approximate message passing with arbitrary matrices," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5339–5351, 2019.

[115] S. Rangan, A. K. Fletcher, P. Schniter, and U. S. Kamilov, "Inference for generalized linear models via alternating directions and Bethe free energy minimization," *IEEE Trans. Inf. Theory*, vol. 63, no. 1, pp. 676–697, 2017.

[116] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing—part i: Derivation," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5839–5853, 2014.

[117] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing part ii: Applications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5854–5867, Nov 2014.

[118] Y. Deshpande and A. Montanari, "Information-theoretically optimal sparse PCA," in *Proc. IEEE Int. Symp. Inf. Theory*, June 2014, pp. 2197–2201.

[119] T. Lesieur, F. Krzakala, and L. Zdeborová, "Phase transitions in sparse PCA," in *Proc. IEEE Int. Symp. Inf. Theory*, 2015, pp. 1635–1639.

[120] Y. Kabashima, F. Krzakala, M. Mézard, A. Sakata, and L. Zdeborová, "Phase transitions and sample complexity in Bayes-optimal matrix factorization," *IEEE Trans. Inf. Theory*, vol. 62, no. 7, pp. 4228–4265, 2016.

[121] S. Rangan and A. K. Fletcher, "Iterative estimation of constrained rank-one matrices in noise," in *Proc. IEEE Int. Symp. Inf. Theory*, 2012, pp. 1246–1250.

[122] R. Matsushita and T. Tanaka, "Low-rank matrix reconstruction and clustering via approximate message passing," in *Advances in Neural Information Processing Systems*, 2013, pp. 917–925.

[123] C. Liang, J. Ma, and L. Ping, "Towards Gaussian capacity, universality and short block length," in *Proc. 9th Int. Symp. Turbo Codes Iterative Inf. Process.*, 2016, pp. 412–416.

[124] S. Liang, J. Ma, and L. Ping, "Clipping can improve the performance of spatially coupled sparse superposition codes," *IEEE Commun. Letters*, vol. 21, no. 12, pp. 2578–2581, Dec. 2017.

[125] C. Liang, J. Ma, and L. Ping, "Compressed FEC codes with spatial-coupling," *IEEE Commun. Letters*, vol. 21, no. 5, pp. 987–990, 2017.

[126] S. Liang, C. Liang, J. Ma, and L. Ping, "Compressed coding, AMP-based decoding, and analog spatial coupling," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7362–7375, 2020.

[127] R. Tanner, "Convolutional codes from quasi-cyclic codes: A link between the theories of block and convolutional codes," University of California, Santa Cruz, Computer Research Laboratory, Tech. Rep., 1987.

[128] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sept. 1999.

[129] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," Jet Propuls. Lab., Pasadena, CA, USA, Tech. Rep., Aug. 2003.

[130] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sept. 2015.

[131] D. J. Costello, L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell, and R. Smarandache, "Spatially coupled sparse codes on graphs: theory and practice," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 168–176, July 2014.

[132] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[133] C. Measson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5277–5307, 2008.

[134] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Wave-like solutions of general 1-D spatially coupled systems," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4117–4157, 2015.

[135] R. El-Khatib, N. Macris, T. Richardson, and R. Urbanke, "Displacement convexity in spatially coupled scalar recursions," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 604–621, 2019.

[136] L. Zdeborová and F. Krzakala, "Statistical physics of inference: thresholds and algorithms," *Advances in Physics*, vol. 65, no. 5, pp. 453–552, 2016.

[137] Y. Wu and S. Verdú, "Rényi information dimension: Fundamental limits of almost lossless analog compression," *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 3721–3748, 2010.

[138] S. Kudekar and H. D. Pfister, "The effect of spatial coupling on compressive sensing," in *Proc. 48th Annual Allerton Conf. Commun., Control, Comput.*, 2010, pp. 347–353.

[139] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7434–7464, Nov. 2013.

[140] J. Barbier, M. Dia, and N. Macris, "Proof of threshold saturation for spatially coupled sparse superposition codes," in *Proc. IEEE Int. Symp. Inf. Theory*, July 2016, pp. 1173–1177.

[141] S. Hamed Hassani, N. Macris, and R. Urbanke, "Threshold saturation in spatially coupled constraint satisfaction problems," *Journal of Statistical Physics*, vol. 150, no. 5, pp. 807–850, 2013.

[142] D. Achlioptas, S. H. Hassani, N. Macris, and R. Urbanke, "Bounds for random constraint satisfaction problems via spatial coupling," in *Proc. Annual ACM-SIAM Symp. Discrete Algorithms*, 2016, pp. 469–479.

[143] V. Aref, N. Macris, and M. Vuffray, "Approaching the rate-distortion limit with spatial coupling, belief propagation, and decimation," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 3954–3979, 2015.

[144] K. Takeuchi, T. Tanaka, and T. Kawabata, "Improvement of BP-based CDMA multiuser detection by spatial coupling," in *Proc. IEEE Int. Symp. Inf. Theory*, 2011, pp. 1489–1493.

[145] A. Yedla, P. S. Nguyen, H. D. Pfister, and K. R. Narayanan, "Universal codes for the Gaussian MAC via spatial coupling," in *Proc. 49th Annual Allerton Conf. Commun., Control, Comput.*, 2011, pp. 1801–1808.

[146] D. Truhachev, "Universal multiple access via spatially coupling data transmission," in *Proc. IEEE Int. Symp. Inf. Theory*, 2013, pp. 1884–1888.

[147] C. Schlegel and D. Truhachev, "Multiple access demodulation in the lifted signal graph with spatial coupling," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2459–2470, 2013.

[148] S. H. Hassani, N. Macris, and R. Urbanke, "Coupled graphical models and their thresholds," in *Proc. IEEE Inf. Theory Workshop*, 2010, pp. 1–5.

[149] S. H. Hassani, N. Macris, and R. Urbanke, "Chains of mean-field models," *J. Stat. Mech.: Theory Exp.*, vol. 2012, no. 02, p. P02011, Feb. 2012.

[150] A. Giurgiu, N. Macris, and R. Urbanke, "Spatial coupling as a proof technique and three applications," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5281–5295, 2016.

[151] "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)," European Telecommunications Standards Institute (ETSI) EN 302 307-1 V1.4.1, 2014.

[152] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[153] R. El-Khatib and N. Macris, "The velocity of the propagating wave for spatially coupled systems with applications to LDPC codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7113–7131, 2018.

[154] A. Cassagne, O. Hartmann, M. Léonardon, T. Tonnellier, G. Delbergue, C. Leroux, R. Tajan, B. Le Gal, C. Jégo, O. Aumage, and D. Barthou, "Fast simulation and prototyping with AFF3CT," in *Proc. IEEE Int. Workshop on Signal Processing Systems*, Oct. 2017.

[155] K. Hsieh, "A Python implementation of sparse regression codes," https://github.com/kuanhsieh/sparc_public, 2020.

[156] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, 2011.

[157] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2277–2292, 2012.

[158] M. Zhu, D. G. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, "Combating error propagation in window decoding of braided convolutional codes," in *Proc. IEEE Int. Symp. Inf. Theory.* IEEE, 2018, pp. 1380–1384.

[159] A. Greig, "Design techniques for efficient sparse regression codes," Ph.D. dissertation, Cambridge University, 2018.

[160] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP)," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4290–4308, July 2013.

[161] L. Anitori, A. Maleki, M. Otten, R. G. Baraniuk, and P. Hoogeboom, "Design and analysis of compressed sensing radar detectors," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 813–827, Feb. 2013.

[162] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of large MIMO detection via approximate message passing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2015.

[163] D. Guo, Y. Wu, S. S. Shitz, and S. Verdú, "Estimation in Gaussian noise: Properties of the minimum mean-square error," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2371–2385, 2011.

[164] D. Guo and S. Verdú, "Randomly spread CDMA: asymptotics via statistical physics," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1983–2010, 2005.

[165] S. Rangan, A. K. Fletcher, and V. K. Goyal, "Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1902–1923, 2012.

[166] A. Bereyhi, R. R. Müller, and H. Schulz-Baldes, "Statistical mechanics of MAP estimation: General replica ansatz," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 7896–7934, 2019.